

Mikko Yletyinen

Langattoman värinämittausverkon mittausvastaanotin

Insinööritö
Kajaanin ammattikorkeakoulu
Tekniikka ja liikenne
Tietotekniikka
2009



**Kajaanin
ammattikorkeakoulu**

OPINNÄYTETYÖ TIIVISTELMÄ

| | |
|---|---|
| Koulutusala Tekniikka ja liikenne | Koulutusohjelma Tietotekniikka |
| Tekijä(t) Mikko Yletyinen | |
| Työn nimi Langattoman tärinämittausverkon mittausvastaanotin | |
| Vaihtoehtoiset ammattiopinnot Sulautetut järjestelmät | Ohjaaja(t) Jukka Heino |
| | Toimeksiantaja Ismo Talus |
| Aika Marraskuu 2009 | Sivumäärä ja liitteet 41 + 19 |
| <p>Insinöörityössä perehdyttiin tärinään, sen etenemiseen maaperässä sekä tärinöiden mittaukseen. Erilaiset rakennustoiminnot, kuten louhinta ja paalutus, aiheuttavat rakenteille vaarallisia tärinöitä. Jotta rakenteiden vaurioitumisvaaraa voidaan arvioida, tärinöitä mitataan erityisillä tärinämittareilla. Tärinämittarit ovat monikanavaisia mitalaitteita, joissa analoginen tärinäsignaali muutetaan digitaaliseksi dataksi tallennusta varten. Rakennustärinöiden mittauksessa on vakiintunut käyttöön heilahdusnopeuden mittaus, joka korreloi parhaiten syntyneiden vaurioiden suuruuden kanssa. Koska rakennustärinöiden mittauksessa ei ole vielä kaupallisesti sovellettu lyhyen kantaman langattomia anturiverkkoja, perehdyttiin insinöörityössä myös ZigBee-standardin mukaisen anturiverkon toimintaan.</p> <p>Insinöörityön tavoitteena oli rakentaa langattoman tärinämittausverkon mittausvastaanotin mittausverkon hallintaan ja mittausdatan lukemiseen antureilta.</p> <p>Työssä rakennettiin ZigBee-mittausverkon päätelaite, jolla voidaan ohjata mittausantureita sekä mittausverkkoa ja lisäksi sillä voidaan lukea verkossa olevien anturien mittaus tietoja. Mittausvastaanotinta ohjaa ATmega 128-mikrokontrolleri. Laitteessa on 128 x 64 pisteen graafinen LCD-näyttö sekä 8-näppäiminen näppäimistö. Tiedon siirtoa varten laitteessa on Meshneticsin valmistama ZigBit-moduuli.</p> <p>Laite toimi rakennetuilta osiltaan odotusten mukaisesti ja laitteen toimintaa ZigBee-verkossa testattiin menestyksekkäästi Meshneticsin WSN Monitor -ohjelmalla, Meshneticsin valmistamien kehityskorttien kanssa.</p> | |
| Kieli | Suomi |
| Asiasanat | Tärinän mittaus, ZigBee, anturiverkko |
| Säilytyspaikka | <input checked="" type="checkbox"/> Ammattikorkeakoulujen Theseus-verkkokirjasto <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto |

| | |
|--|---|
| School School of Engineering | Degree Programme Information Technology |
| Author(s) Mikko Yletyinen | |
| Title A Transceiver for a Wireless Vibration Measurement Network | |
| Optional Professional Studies Embedded Systems | Instructor(s) Mr Jukka Heino |
| | Commissioned by Mr Ismo Talus |
| Date November 2009 | Total Number of Pages and Appendices 41+19 |
| <p>This Bachelor's thesis is about vibration, its physical properties as a wave and the measurement of it. Vibration on the earth is composed of four different formations of waves: the P and S waves as the basic element of waves in an earthquake, for instance. The Rayleigh and Love waves are more or less mixtures of P and S waves. All four types of waves have a different appearance and way of movement in soil. The effect on structures also varies, the Rayleigh and Love waves being responsible for the most of the damages produced.</p> <p>The measurement of vibration in soil is conducted with purpose-built measurement devices. These devices usually have 4-6 measurement channels, a digitizer and embedded memory for the storage of measurement data. The price of these devices ranges from 6 000 to 10 000 euros.</p> <p>The ZigBee standard defines a protocol for short range wireless networking. ZigBee is indented for use in sensor systems and control signaling within the range up to 100 meters and comprising data throughput up to 250 Kbps.</p> <p>The initial idea of the thesis was to develop a wireless measurement network for measuring vibration in buildings during rock blasting or piling operations. The system consists of the controlling unit attachable to a PC and several wireless vibration sensors.</p> <p>There are growing markets for sensor networks and cheap sensors, and there is no similar existing system in the markets currently. The initial idea being too large for a thesis, it was narrowed to comprise the development of a controlling unit for the measurement network.</p> <p>The controlling unit was designed to operate under the Atmel's ATMega 128 microcontroller and the ZigBee networking features were designed to be utilized by the Meshnetics ZigBit module and Bitcloud protocol stack.</p> | |
| Language of Thesis Finnish | |
| Keywords | ZigBee, wireless sensor network, vibration measurement |
| Deposited at | <input checked="" type="checkbox"/> Theseus Electronic library <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences |

ALKUSANAT

Seuraavat, Kajaanin ammattikorkeakoulun henkilökuntaan kuuluvat, ansaitsevat kiitokset avustaan tämän insinöörityön tekemisessä ja loppuunsaattamisessa:

Markku Karppinen, Jukka Heino, Eero Soininen, Kaisu Korhonen, Ismo Talus, Arto Partanen ja Harri Honkanen

Kiitos myös perheelleni jaksamisesta ja ymmärtämisestä.

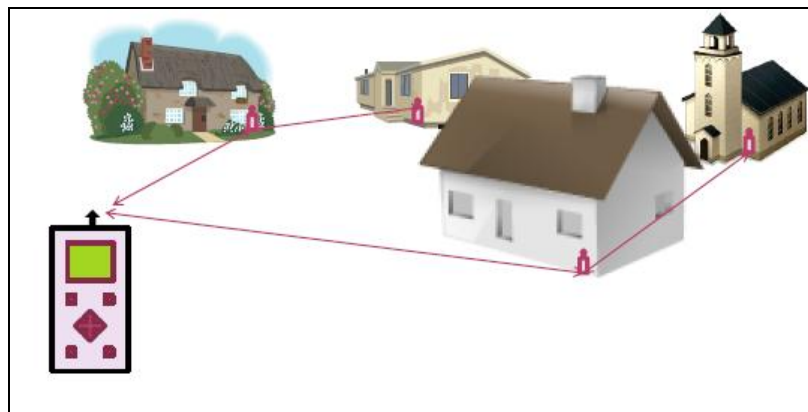
SISÄLLYS

| | |
|--|----|
| 1 JOHDANTO | 1 |
| 2 TÄRINÄ, SEN ETENEMINEN MAAPERÄSSÄ JA TÄRINÄN MITTAAMINEN | 3 |
| 2.1 Tärinä fysikaalisena ilmiönä | 3 |
| 2.2 Aaltoliikkeen peruskäsitteitä | 3 |
| 2.3 Tärinän eteneminen maaperässä [3] | 7 |
| 2.4 Tärinän vaikutukset rakenteisiin | 9 |
| 2.5 Tärinän mittaaminen | 11 |
| 2.5.1 Mittausmenetelmät | 12 |
| 2.5.2 Mittalaitteet | 14 |
| 3 LYHYEN KANTAMAN SENSORIVERKKO – ZIGBEE | 17 |
| 3.1 ZigBee-protokolla | 17 |
| 3.1.1 ZigBee-verkon rakenne | 20 |
| 3.1.2 ZigBeen käyttösovellukset | 21 |
| 3.2 Meshnetics ZigBit | 22 |
| 4 LANGATTOMAN TÄRINÄMITTARIN MITTAUSVASTAANOTIN | 25 |
| 4.1 Suunnittelu | 25 |
| 4.1.1 Vaatimusten määrittely | 25 |
| 4.1.2 Laitteen suunnittelu | 26 |
| 4.1.3 Kytkennän ja piirilevyn suunnittelu | 32 |
| 4.1.4 Ohjelmiston suunnittelu | 33 |
| 4.2 Laitteen rakentaminen | 33 |
| 4.3 Laitteen ohjelmointi ja testaus | 35 |
| 5 TYÖN TAVOITTEIDEN SAAVUTTAMINEN JA LANGATTOMAN TÄRINÄMITTARIN JATKOKEHITYS | 37 |
| 5.1 Työn tulokset | 37 |
| 5.2 Jatkokehitys | 37 |
| 6 YHTEENVETO | 39 |
| LÄHTEET | 40 |
| LIITTEET | |

1 JOHDANTO

Idean tähän insinöörityöhön olen saanut perehtyessäni tärinänmittauksiin louhintatyömailla. Useassa louhintakohteessa tarvitaan tärinänmittausta varmistamaan, ettei ympäröiville taloille, rakenteille tai herkille laitteille aiheudu vaurioita räjäytysten aikaansaamista tärinöistä. Nykyään käytössä olevat tärinänmittauslaitteet ovat kohtalaisen kalliita, mittariin liitettävien anturien määrä on rajallinen sekä anturien liittäminen mittariin tapahtuu yleensä kaapelin avulla.

Noista seikoista johtuen sain idean järjestelmästä, jolla tärinät voitaisiin mitata langattomasti yhdestä paikasta ja sen lisäksi mittauksen suorittaisivat yksinkertaiset anturit ja anturien mitaamat tärinät luettaisiin yhdellä päätelaitteella (kuva 1).



Kuva 1. Langaton tärinämittari

Alkuperäinen idea jalostui eräänlaiseksi tärinänmittausverkoksi, kun tutustuin lyhyen kantan ZigBee-anturiverkkojen rakenteeseen ja ominaisuuksiin.

Työn kestäessä insinöörityöaihetta rajattiin siten, että varsinaiseksi työksi tuli ZigBee-anturiverkon mittausvastaanottimen rakentaminen. Mikäli olisin halunnut rakentaa toimivan järjestelmän, olisi mittausvastaanottimen lisäksi pitänyt rakentaa anturiosia, mikä olisi ollut tämän työn aikataulun puitteissa mahdoton tai ainakin äärimmäisen vaikea tehtävä. Kuitenkin työn testauksessa käytettiin ”anturiosana” Meshneticsin ZigBit-kehityspaketin laitteita, joissa on valmiina mm. lämpötila- ja valoanturit. Näin ollen mittausvastaanottimen rakentaminen oli sikäli mielekästä, että oli jo olemassa anturit, joiden mittausdataa voitiin vastaanottaa vaikkakaan se ei ole tärinää.

Käsittelen ensin tärinää fysikaalisena ilmiönä sekä sitä, miten tärinä etenee maaperässä. Lisäksi työssä käsitellään hieman tärinän mittauksen periaatteita. Tarkoitus on antaa yleiskuvauks siitä, mitä on tarkoitus mitata, millaisin menetelmin mittausta tehdään sekä miten mittausdata siirretään. Lisäksi käsitellään ZigBee-standardia yleisesti ja kerrotaan langattoman antenniverkon yleisiä ominaisuuksia.

Toisessa osassa selvitän työssä rakennetun mittausvastaanottimen rakennetta, rakentamista ja testaamista sekä esitän mahdollisuuksia, miten laitetta voidaan kehittää.

2 TÄRINÄ, SEN ETENEMINEN MAAPERÄSSÄ JA TÄRINÄN MITTAAMINEN

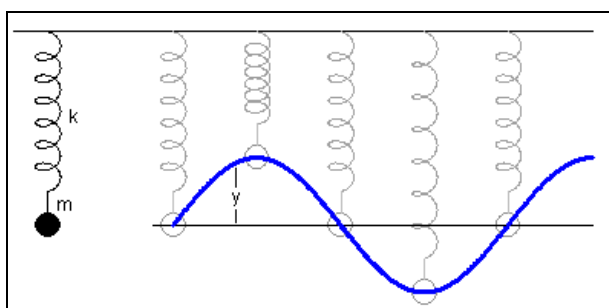
2.1 Tärinä fysikaalisena ilmiönä

Tärinä on fyysisen voiman ilmentymä. Tärinä ei ole yksittäinen tapahtuma, pulssi, vaan se on ajallisesti tietyn aikaa kestävä ilmiö. Tärinästä voidaan eritellä ajanhetkiä, jolloin fyysinen voima on kovimmillaan, heikoimmillaan tai nolla. Lisäksi fyysinen voima voi tärinän aikana vaihtaa suuntaa. Noista seikoista voidaan tehdä päätelmä, että tärinä on jonkinlaista aaltoliikettä.

Aaltoliike jaetaan yleisesti pitkittäiseen ja poikittaiseen aaltoliikkeeseen. Aaltoliike voi olla myös sekamuotoista, kuten esimerkiksi elliptiset tärinääallot, jotka etenevät maankuoren pinnassa maanjäristyksen aikana.

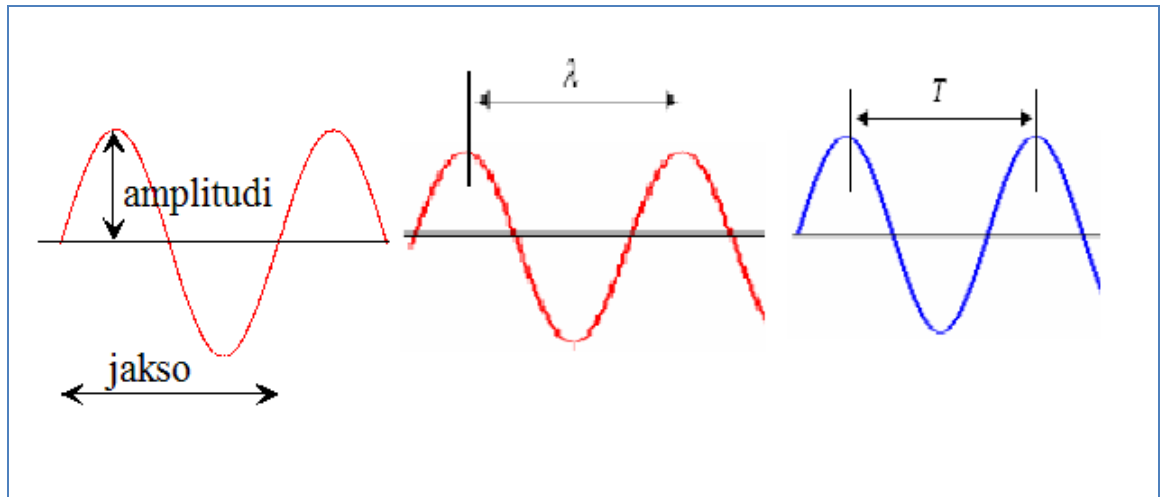
2.2 Aaltoliikkeen peruskäsitteitä

Kuva 4. esittää punnusta jousen päässä. Kun punnusta venytetään alaspäin ja päästetään siitä irti, alkaa jousi-punnus-systeemi värähdellä edestakaisin. Värähtelyn aikaansaa punnukseen alaspäin vaikuttava maan vetovoima ja ylöspäin vaikuttava jousen jousivoima. Reaalimaailmassa kuvan systeemin värähtely vaimenee hyvin nopeasti jousen muodonmuutoksessa ilmenevän kitkan sekä ilmanvastuksen takia. Reaalimaailmassa puhutaan vaimenevasta värähdysliikkeestä. Teoriassa voidaan ajatella värähtelyn jatkuvan äärettömyyteen, jolloin punnuksen paikka ajan hetkellä (t) piirtää jatkuvan sinimuotoisen kuvaajan.



Kuva 4. Jousiheiluri [1]

Perinteisin aaltoliikkeen kuvaaja on kuvan 5 tapausten mukainen. Kuvassa on esitetty lyhyitä ”pätkiä” sinimuotoisesta signaalista. Kuvan vasemmanpuoleisimmassa kuvaajassa on esitetty amplitudin ja jakson käsite. Aaltoliikkeen huippujen eli amplitudien väliä kutsutaan aallonpituudeksi, λ . (Kuvassa 5, keskimmäinen kuvaaja). Koska aaltoliike on nimensä mukaisesti liikettä, on mukana oltava myös ajallinen komponentti. Aikaa, joka kuluu amplitudien välillä, kutsutaan jaksonajaksi T (kuva 5, oikea kuvaaja) Jaksonajan yksikkö on sekunti (s).



Kuva 5. Siniaallon käsitteitä

Aaltoliikkeen taajuus kuvaa värähtelyn nopeutta. Mitä suurempi on taajuus, sitä nopeammin esimerkiksi sinimuotoisen aaltoliikkeen amplitudit esiintyvät. Taajuudella ja jaksonajalla on selvä yhteys; eli taajuus (f) voidaan ilmoittaa jaksonajan avulla seuraavasti:

$$f = 1/T \quad (1)$$

Eli taajuus on jaksonajan käänteisarvo. Taajuuden yksikköä $1/s$ kutsutaan hertsiksi (Hz).

Taajuudesta sijasta voidaan käyttää myös termiä kulmataajuus (tai kulmanopeus).

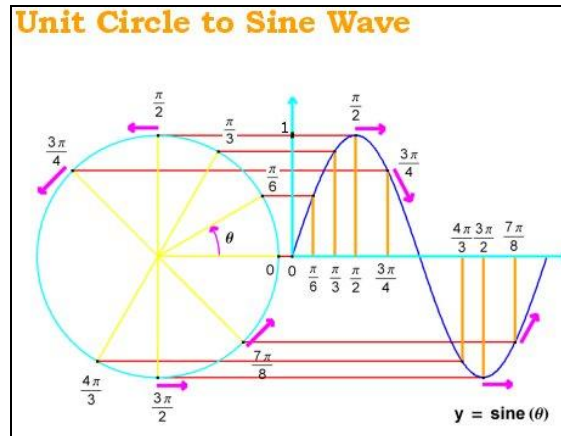
Taajuuden ja kulmataajuuden yhteys on seuraava:

$$\text{Kulmataajuus, } \omega = 2\pi f \quad (2)$$

kulmataajuuden yksikkö on rad/s.

Kun puhutaan kulmataajuudesta (tai -nopeudesta), käytetään jaksollisen aaltoliikkeen esittämiseen hyväksi vektoria, jonka suuruus ja suunta vaihtelevat siten, että vektorin 0-piste on

ympyrän keskipisteessä ja vektorin päätepiste ympyrän kehällä (kuva 6.). Tällöin kulmaa vektorin ja vaaka-akselin välillä voidaan kutsua vaihekulmaksi. Vaihekulman muuttumisnopeus on kulmanopeus.



Kuva 6. Siniaalto yksikköympyrän avulla

Aaltoliikkeen nopeus saadaan johdettua em. suureiden perusteella. Nopeudestahan (v) tiedämme, että se on yhtä kuin liikuttu matka (s) jaettuna matkaan kuluneella ajalla (t), eli:

$v = s/t$, joten sijoittamalla suureet saadaan aaltoliikkeen nopeudeksi (V):

$$V = \lambda/T, \text{ josta edelleen } V = \lambda f \quad (3)$$

Tämä suure kuvaa siis aallon etenemisnopeutta.

Ns. harmonisen aaltoliikkeen funktion kuvaaja syntyy funktiosta:

$$U(t) = A_0 \sin \omega t, \quad (4)$$

missä A_0 = aallon maksimi amplitudi ja ω = kulmanopeus ($=2\pi f$)

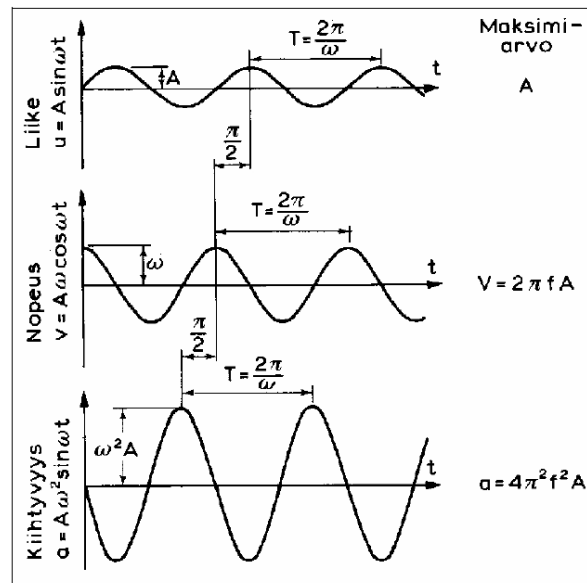
Edellisestä kaavasta 4. derivoimalla saadaan värähtelijän heilahdusnopeus kunakin ajan hetkenä:

$$V(t) = A_0 \omega \cos \omega t \quad (5)$$

Ja edelleen nopeuden kaavasta (5.) derivoimalla ajan suhteen saadaan kiihtyvyys:

$$a(t) = -A_0 \omega^2 \sin \omega t \quad (6)$$

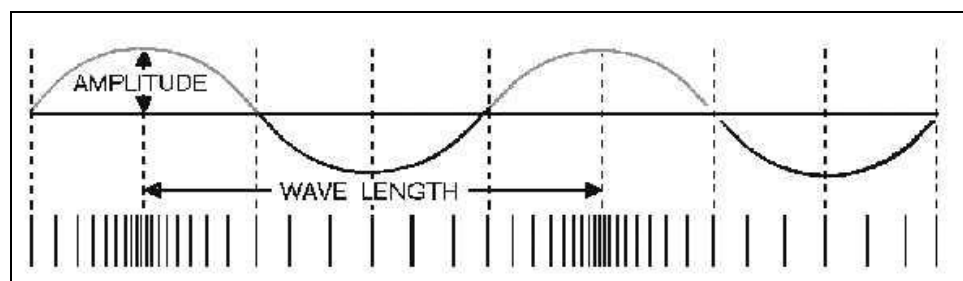
Kuten kuvasta 7 voidaan havaita, on nopeuden vaihe $\pi/2$ verran paikan vaihetta jäljessä, kuten myös kiihtyvyyden vaihe on $\pi/2$ verran nopeutta jäljessä. Jos ajatellaan jousen tapausta, niin tuo vaihe-ero tuntuu järkeenkäyvältä. Kun jousi-punnus-systeemi on tasapainokohdassaan (poikkeama $A=0$), on kiihtyvyys nolla. Eli juuri tässä kohdassa alaspäin vaikuttava maan vetovoima ja ylöspäin vaikuttava jousivoima ovat yhtä suuret ja siten kumoavat toisensa. Samoissa 0-kohdissa myös nopeus on suurimmillaan.



Kuva 7. Aaltoliikkeen poikkeama, nopeus ja kiihtyvyys [2, s. 164]

Edeltävässä on käsitelty pelkästään poikittaista aaltoliikettä. Aaltoliikettä on myös pitkittäistä.

Kuvassa 8 on esitetty analogia pitkittäisen ja poikittaisen aaltoliikkeen välillä. Pitkittäinen aaltoliike muodostuu tihentymistä, joita edustavat poikittaisen aaltoliikkeen positiiviset huiput sekä harventumista, joita edustavat poikittaisen aaltoliikkeen negatiiviset huiput.



Kuva 8. Poikittainen ja pitkittäinen aaltoliike [25]

Luonnossa esiintyy molempia aaltoliikkeiden muotoja. Esimerkiksi ääni on pitkittäistä aaltoliikettä ja koostuu ilmamolekyylien tihtymistä ja harventumista, ts. ilman paineen vaihtelusta. Poikittaista aaltoliikettä luonnossa on esimerkiksi valo, jolla on tosin omat erikoispiirteensä kuten sähkö- ja magneettikentän duaalisuus sekä kyky edetä ilman väliainetta.

Yksi huomionarvoinen seikka aaltoliikkeestä on se, että aaltoliike ei siirrä materiaa. Ts. väliaineen hiukkaset siirtyvät paikaltaan mutta palaavat keskimäärin takaisin samalle paikalleen, kuin ennen aaltoliikkeen esiintymistä. Mitä aaltoliike sitten siirtää? – Aaltoliike siirtää energiaa.

2.3 Tärinän eteneminen maaperässä [3]

Aaltoliikkeen etenemistä maaperässä tutkii tieteenala nimeltä seismologia. Sana seismologia on johdettu kreikan kielen sanasta seismos, joka tarkoittaa maanjäristystä, ja sanasta logos, joka tarkoittaa tietoa. Eli seismologia siis tutkii tietoa maanjäristyksistä.

Aaltoliikettä maankuoressa sanotaan seismisiksi aalloiksi, siis maanjäristys-aalloiksi. Seismiset aallot eivät tosin etene pelkästään maankuoressa, ne voivat edetä jopa maapallon läpi. Tosin seismisistä aalloista ainoastaan P-aallot eli pitkittäinen aaltoliike voi edetä maapallon nestemäisen ulkoytimen läpi.

Tärinä maaperässä syntyy jonkin fyysisen voiman nopeassa purkautumisessa. Se voi olla mannerlaattojen törmäyskohdassa vuosien aikana syntyneen jännityksen purkautuminen, jolloin syntyy maanjäristys. Myös maaperässä räjähtävä räjähdyspanos, rakennuspaalutus tai vaikkapa raideliikenne aikaansaa samantyyppiset tärinääallot. Olipa tärinän synty tapa millainen tahansa, syntyneet tärinääallot voidaan luokitella seuraavasti:

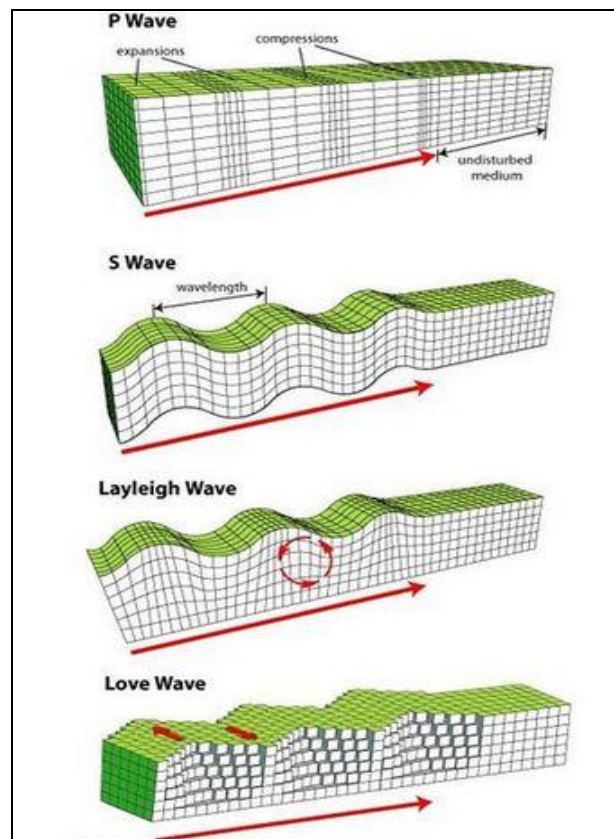
1. Välittömästi voiman vaikutuspisteestä pallomaisesti kaikkiin suuntiin etenevä primääriaalto, P-aalto, joka on tärinääalloista kaikkein nopein. Primääri-aaltoliike on pitkittäistä aaltoliikettä, joka etenee graniitissa noin 6 km/s. P-aalto aiheuttaa väliaineeseen puristusta ja venytystä.
2. Niin ikään kaikkiin suuntiin etenevä sekundaariaalto, S-aalto, jonka nopeus kalliossa on noin 60 % P-aallon nopeudesta. S-aalto on poikittaista aaltoliikettä ja aiheuttaa väliaineen hiukkasissa ylös - alas liikettä. S-aalto aiheuttaa esimerkiksi rakenteisiin leik-

kauksia ja siksi joissakin englanninkielisissä lähteissä sitä kutsutaan Shear-wave-nimellä.

3. Pinta-aallot syntyvät, kun ensin P-aalto ja myöhemmin S-aalto etenee maan pintaan, heijastuu ja taittuu. Tällöin osa aaltoliikkeestä lähtee etenemään pinta-aaltona maankuoren pintaosissa. Pinta-aaltoja ovat Rayleigh-aallot (R-aalto) sekä Love-aalto (L-aalto). Ne ovat saaneet nimensä keksijöidensä mukaan. R-aallossa aaltomuoto on ellipsimäinen, sisältäen molemmat ominaisuudet P- ja S-aallosta. R-aalto aiheuttaa väliaineen leikkausta ja puristumista. Materiaalipartikkelien liike tapahtuu vertikaalisuunnassa.
4. L-aalto, jota kutsutaan myös Q-aalloksi, on luonteeltaan samankaltainen kuin S-aalto. L-aallossa kuitenkin partikkelien liike on horisontaalitasossa.

P- ja S-aallot ovat ns. runkoaaltoja, vaikkakin näiden kuljettama energiamäärä on suhteessa pieni. L- ja R-aallot kuljettavat enemmän energiaa ja siksi, esimerkiksi maanjäristyksessä, saavat aikaan suurimmat tuhot.

Kuvassa 9 on esitetty värinäaaltojen ominaisuuksia. Kaikkein nopeimmin etenevä P-aalto aiheuttaa maaperässä puristus- ja venytysjännityksiä sekä partikkeleiden liikettä edestakaisin aallon kulkusuunnassa. S-aallossa partikkelit liikkuvat aallon etenemissuuntaan nähden kohtisuorasti, ylös-alas. Maanjäristyksen keskusta eli fokusta paikannettaessa käytetään hyväksi P- ja S-aaltojen kulkuaikojen eroa.



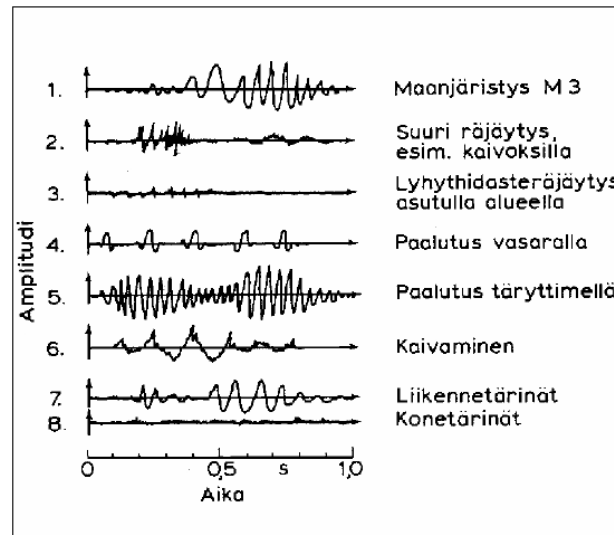
Kuva 9. Erityyppiset värinäaallot maankuoressa [4]

2.4 Tärinän vaikutukset rakenteisiin

Pohjois-Eurooppa sijaitsee maanjäristyksen syntypaikkoja ajatellen jokseenkin rauhallisella alueella. Suomessakin tosin esiintyy maanjäristyksiä, jotka ilmeisesti johtuvat maankuoren kohoamisesta. Eniten Suomessa tapahtuu maanjäristyksiä Perämeri–Kuusamo-välisellä vyöhykkeellä [5, s. 35]. Historian kirjoista löytyy merkintöjä kyllä tuhoisistakin maanjäristyksistä. Vuonna 1626 on Pohjanmaalla ja Kainuussa tapahtunut maanjäristys, jossa mm. Paltamon kirkko Paltaniemellä on vaurioitunut pahoin ja jouduttiin myöhemmin rakentamaan uudelleen [6.]

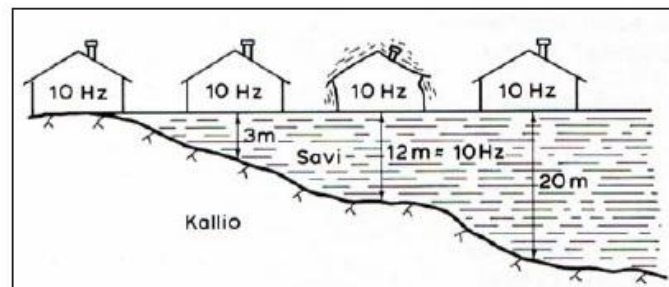
Vaikka Suomessa maanjäristykset ovat harvinaisia, niin maaperän tärtelyjä seurataan aktiivisesti aivan muiden seikkojen takia. Kuten maanjäristys, saattavat myös erilaiset ihmisen toiminnan aikaansaamat tärinät vaurioittaa mm. rakennuksia tai tärinäherkkiä laitteita.

Tärinöitä syntyy pääasiassa rakentamistoiminnoista; kallion louhinta, paalutus ja niin edelleen. Kuvassa 10 on esitetty erilaisten tärinälähteiden aikaansaamia amplitudikuvaajia.



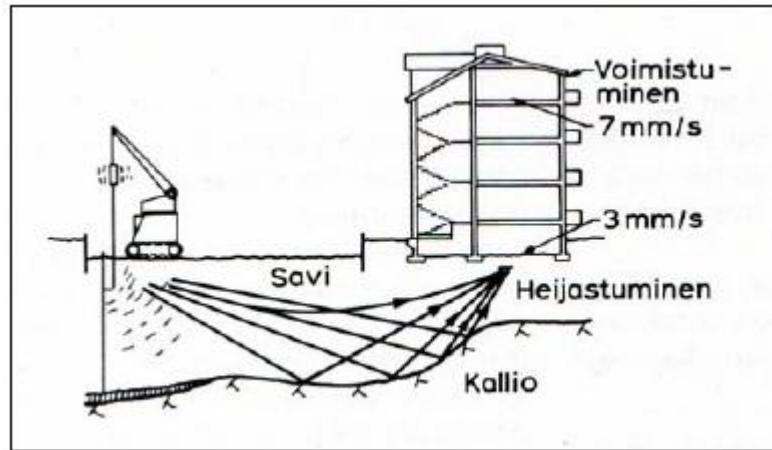
Kuva 10. Erilaisia rakentamisen aiheuttamia tärinöitä [2, s. 164]

Ihminen tuntee pienenkin tärinän ”herkemmin” kuin mikään mittalaite. Tällöin ihmiset ovat monesti huolissaan talojensa rakenteiden kestävydestä, kun läheisyydessä tehdään esimerkiksi räjäytystöitä. Toki on totta, että varomaton toiminta saattaa aikaansaada vahinkoja rakenteissa. Tärinäaallot voivat heijastua kallion ja saven rajapinnasta sekä summautua, jolloin tärinä saattaa esiintyä yllättävän voimakkaana, kuten kuvan 11 tilanteessa.



Kuva 11. Tärinä voi esiintyä yllättävästi [2, s. 167]

Rakennuksella on myös tietty ominaisvärähtelytaajuus. eli taajuus, jolla rakennus värähtelee helpoiten. Tällöin saattaa käydä niin, että, kun systeemiin (=talo) syötetään taajuudeltaan ominaisvärähtelytaajuudella olevaa tärinää, talo alkaa itse värähdellä tuolla taajuudella ja systeemiin (=talo) syötetty jatkuva tärinä vahvistaa värähtelyä entisestään, jolloin talon rakenteeseen syntyy vaurioita (kuva 12). Esimerkiksi siltoja tiedetään sortuneen tuulen aikaansaaman resonanssin vaikutuksesta.



Kuva 12. Resonanssin aikaansaama tärinän voimistuminen [2, s. 167]

Länsimaisessa oikeuskäytännössä tekijä on vastuussa vahingoista. Suomessa esimerkiksi louhinta on ns. ankaran vastuun alaista toimintaa, jolloin vauriotapauksessa louhintaurakoitsijalla on näyttövastuu siitä, ettei hänen toimintansa ole aiheuttanut vaurioita. Niinpä esimerkiksi louhittaessa rakennetulla alueella ovat erilaiset rakennuskatselmukset ja tärinänmittaukset enemmän sääntö kuin poikkeus.

2.5 Tärinän mittaaminen

Tärinänmittauksissa, kuten esimerkiksi teollisuuden kunnossapidon mittauksissa, mitataan yleensä tärinän kiihtyvyyttä. Mitattaessa tärinöitä rakenteista on mitattavaksi suureeksi kuitenkin vakiintunut heilahdusnopeuden mittaaminen. Tämä lienee perua 1940-luvun alkupuolelta ruotsalaisista tärinänmittaustutkimuksista, joissa todettiin heilahdusnopeuden olevan suoraan verrannollinen graniitille rakennettujen rakennusten vaurioitumisissa mitatulla taajuusalueella. Myöhemmät sadattuhannet mittaustulokset vahvistavat ruotsalaisten mittaukset. [2, s. 168.]

Rakenteiden vaurioitumisvaaraa arvioitaessa on tietenkin oleellista tarkastella heilahdusnopeuden maksimi-arvoja. Nopeuden maksimi-arvo voidaan laskea kaavasta:

$$V_{\max} = 2\pi f A_0 \quad (7)$$

Kaavasta 7. voidaan todeta, että heilahdusnopeus on riippuvainen maksimi-amplitudista ja värinääallon taajuudesta.

Vaikka värinääallot esiintyvät monessa eri muodossa, voidaan kuitenkin esimerkiksi räjähdysvärinää pitää siniaaltona. [2, s. 165.] Tai ainakin niiden käsittelyssä on tähän asti niin oletettu.

Reaalimaailmassa jo pelkän seismografin piirtämän kuvaajan muodon perusteella voidaan sanoa, että värinä, riippuen lähteestä, sisältää monia taajuuksia. Mitatusta signaalista voidaan saada selville signaalin spektri tekemällä signaalille Fourierin muunnos.

Ranskalainen matemaatikko Joseph Fourier kehitti 1800-luvun alussa teorian, jonka mukaan jokainen signaali voidaan esittää jaksollisten sinifunktioiden summana eli Fourierin sarjana. Näin ollen jokaisesta signaalista voidaan selvittää sen spektri, eli selvittää signaalissa esiintyvien sinikomponenttien amplitudien voimakkuus. Signaalin spektri selvitetään tekemällä signaalille matemaattisia laskutoimituksia; ns. Fourierin muunnos. Fourierin muunnos on erittäin työläs menetelmä, josta on kuitenkin tullut hyvin yleinen työkalu kiitos nopeiden DSP - prosessorien.

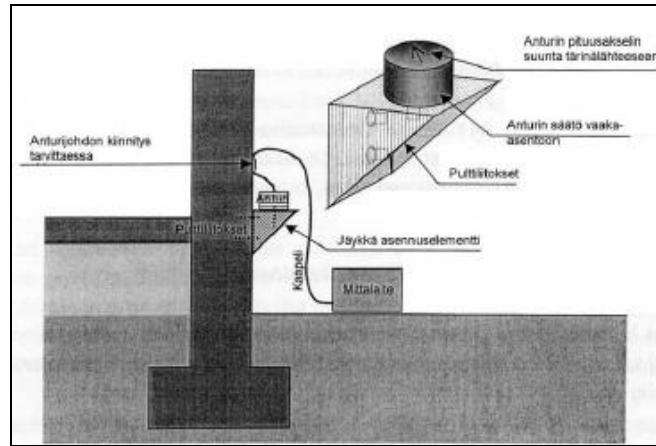
”Värinäsignaaleissa” signaalien taajuudet painottuvat mataliin taajuuksiin ja värinämittareilla yleisesti mitataan taajuuksialueella 2–250 Hz [7].

2.5.1 Mittausmenetelmät

Kansainvälinen ISO-standardointiorganisaatio on laatinut standardin, ISO 4866, jossa annetaan suuntaviivoja värinöiden mittaamiseen rakennuksista sekä mittaustulosten tarkasteluun.

Mitattaessa rakennetta kuormittavaa värinää on mittaavan anturin asennukseen kiinnitettävä huomiota. Anturi on kiinnitettävä lähelle maantasoa paikkaan, josta värinä oletettavasti siirtyy rakenteeseen. Kiinnitys tulee tehdä jäykkään rakenteeseen, kuten esimerkiksi kantavaan sei-

nään tai pilariin (kuva 13). Samoin tulee anturin kiinnityksen olla mahdollisimman jäykkä, jotta värinä pääsee välittymään anturille. Lisäksi anturi tulee kohdistaa oikein; ts. anturin mitausakselit tulee suunnata haluttujen mittaussuuntien mukaiseksi [8, s. 3.]



Kuva 13. Värinäanturin kiinnitys rakennuksen perustukseen [8, s. 4]

Värinöitä voidaan mitata myös maaperästä. Kuvassa 14 on Tampereen teknilliselle yliopistolle diplomityön paalutustärinästä tehneen Hanna Leppäsen mittauksissa käytetty eräs mittalaite. Kuvassa on esitetty kulmaraudasta rakennettu yksinkertainen anturiteline, teräsvaaja, johon on magneeteilla kiinnitetty kolme VTI:n SCA 610-kiikkyvyysanturia. Anturien data on johdettu tiedonkeruuyksikön kautta tietokoneelle. Mittausohjelmistona on käytetty DasyLab-ohjelmaa [9, s. 98.]



Kuva 14. Värinänmittausanturin kiinnitys moreenimaahan [9, s. 98]

Edellisessä esimerkissä oli käytetty mittausjärjestelmää, joka oli rakennettu komponenteista ja mittauksien tulokset käsiteltiin PC:llä. Markkinoilla on kuitenkin ollut jo vuosia erityisesti ra-

kennustärinöiden mittauksiin rakennettuja mittalaitteita, joita yleisesti kutsutaan tärinämittareiksi.

2.5.2 Mittalaitteet

Suurimpia rakennustärinöiden mittaukseen tarkoitettujen tärinämittareiden valmistajia ovat kanadalainen Instandel, ruotsalainen ABEM, sekä amerikkalainen White industrial seismologic inc.

Instandel: [<http://www.instantel.com/>]

ABEM: [<http://www.abem.se/>]

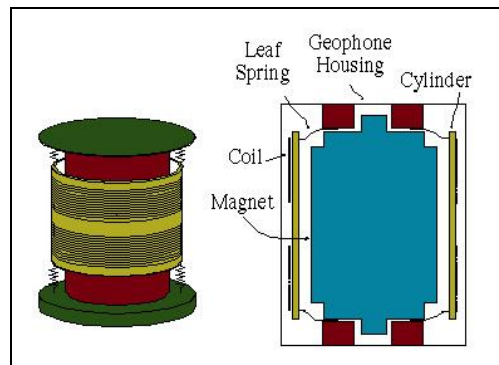
White Industrial seismologic: [<http://www.whiteseis.com/>]

Kullakin valmistajalla on tarjolla useampaa eritasoista laitetta tai mittausjärjestelmää. Tyypillistä näille laitteille on, että ne on suunniteltu kestävänsä ”ulkokäyttöä”. Suomessa käyttäjät yleensä vielä suojaavat laitteet lisäsuojin, kuten esimerkiksi kuvassa 15 on tehty.



Kuva 15. Kolme tärinäanturia ja kolme mittaria ”säänkestävissä koteloidissa” [10]

Mittareissa käytettyjä antureita kutsutaan yleisesti geofoneiksi. Geofoni koostuu yleensä magneetista ja kelasta sekä näihin liitettävästä oheiselektroniikasta, kuten vahvistimista ja suotimista. Tärinän vaikutuksesta kela liikkuu suhteessa magneettiin. Liike indusoi kelan johtimeen jännitteen, joka on verrannollinen liikkeen suuruuteen. Jännite mitataan, vahvistetaan ja suodatetaan. Tämän jälkeen anturin signaali muunnetaan digitaalseksi ja tallennetaan. Kuvassa 16 on esitetty ”perinteisen” geofonin periaatteellinen rakenne.



Kuva 16. Geofonin periaatteellinen rakenne [11]

Nykyään antureiden suunnittelussa ollaan yhä enemmän menossa kohti MEMS-anturien käyttöä.

Tärinämittareissa on yleensä 2-8 mittauskanavaa, ja niihin voidaan liittää tärinäanturien lisäksi joissain tapauksissa myös äänenpaineen mittaus. Kuvassa 17 on esitetty erään valmistajan tärinämittari. Siinä on kuusi mittauskanavaa ja kullekin kanavalle voi valita näytteenottotaajuudeksi 512, 1026, 2048 tai 4096 näytettä/s. Laitteessa on 64 MB muistia mittausten tallentamiseen, ja esimerkiksi näytteenottotaajuudella 1026 S/s laite voi tallentaa noin 7100 kpl yhden sekunnin pituista mittaustapahtumaa kahdella geofonilla. [12.]



Kuva 17. Instantel Minimate Pro6, kuusikanavainen tärinämittari [12]

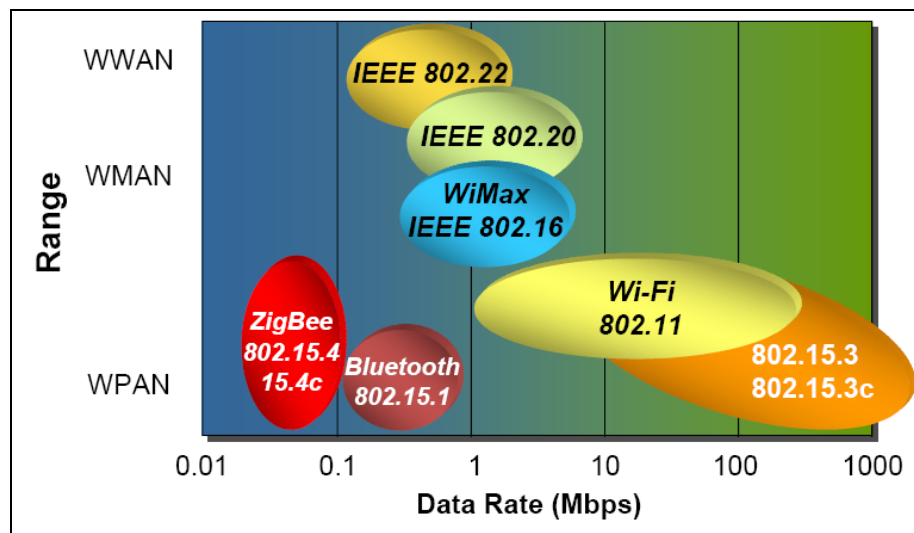
Kuvan laitteen hinta ei ole vielä tiedossa, koska laitetta ei ole vielä Suomessa saatavana. Samaa sarjaan kuuluvan, hieman vanhemman 4-kanavaisen laitteen hinta on maahantuojan mukaan noin 10000 euron luokkaa.[13.]

Monissa laitteissa on kaukovalvontaominaisuus, mikä tarkoittaa sitä, että laite voi lähettää tärinänmittaustiedot GSM/GPRS-verkon kautta mittauspalvelimelle. Varsinaisia langattomia antureita ei vielä ole saatavana, mutta todennäköisesti niitä kuitenkin kehitellään.

Keskustelin puhelimitse erään mittauskonsultin kanssa, jonka mielipide oli, että koska mittarit ovat kalliita, niin tuskin niitä hankitaan lisää. Kuitenkin hän oli sitä mieltä, että mittauspis- teitä voisi olla kyllä enemmänkin. Mielipide puoltasi tarvetta kehittää mittausjärjestelmä, joka koostuu (kalliista) päätelaitteesta, mutta johon voisi kytkeä nykyistä enemmän halpoja antu- reita.

3 LYHYEN KANTAMAN SENSORIVERKKO – ZIGBEE

Lyhyen kantaman langaton verkkotekniikka ZigBee on tarkoitettu hyvin vähän virtaa kuluttavien laitteiden langattomaan tiedonsiirtoon. Yhteysetäisyydet ZigBee-tekniikalla ovat muutamista kymmenistä metreistä noin sataan metriin asti (ulkoisella antennilla jopa yli). Tiedonsiirtokapasiteetti kahden ZigBee-laitteen välillä voi olla 20–250 kb/s. ZigBee-protokollapino vie vähän muistia, jolloin sen käyttöalustat ovat 8-bittisistä mikrokontrollereista ylöspäin. Käyttösovellukset ovat erilaisten laitteiden ohjaussignaalien välitys sekä vaikkapa teollisuuden anturien tiedonsiirto. Kuvassa 18 on esitetty erilaisten langattomien teknologioiden tiedonsiirtokapasiteetit kantamaan suhteutettuna.



Kuva 18. Langattomien teknologioiden vertailu [14]

3.1 ZigBee-protokolla

ZigBee Alliance on useiden piiri- ja laitevalmistajien yhteinen organisaatio, jonka tavoitteena on kehittää protokolla lyhyen kantaman langattomaan tiedonsiirtoon. Yrityksiä on liittynyt ZigBee-allianssiin jo yli 300. ZigBee-protokolla pohjautuu IEEE (Institute of Electrical and

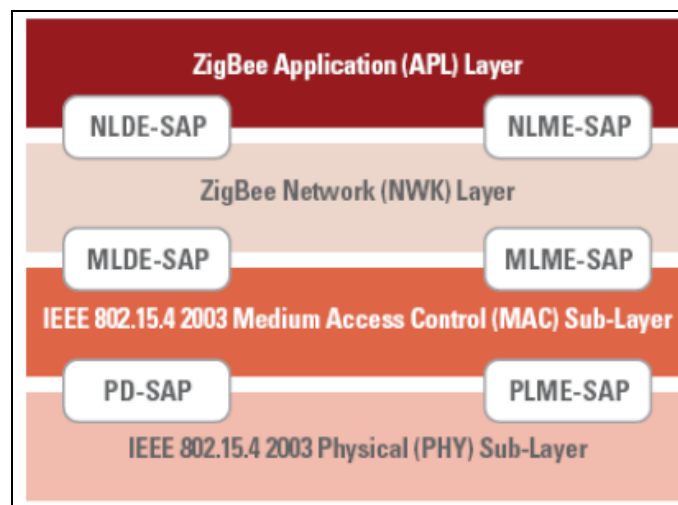
Electronics Engineers)-standardiin 802.15.4. ZigBee-allianssin tavoitteena on ollut luoda edullinen, energiatehokas ja kaksisuuntainen langaton tiedonsiirtostandardi.

ZigBee-allianssin asiakirjassa ZigBee specifications on selvitetty ZigBee-protokollan rakenne ja toiminta yli 600-sivuisessa dokumentissa. Ensimmäinen spesifikaatio on tullut julkiseksi tammikuussa 2004. Eri versiot on numeroitu, ja tätä kirjoittaessa lähdetietoina on käytetty 17. julkaistun version tietoja. [15.]

802.15.4 määrittelee lyhyen kantaman ns. WPAN (Wireless Personal Area Network) protokollan kaksi alinta kerrosta; 1.) MAC eli Medium Access Control sub layer (siirtoyhteyskerros) sekä 2.) PHY eli Physical layer (fyysisen kerroksen). [14, s. 27]

ZigBee-allianssin työnä protokollaan on rakennettu kaksi ylintä kerrosta: NWK eli ZigBee network layer (verkkokerros) sekä APL eli ZigBee application layer (sovelluskerros)

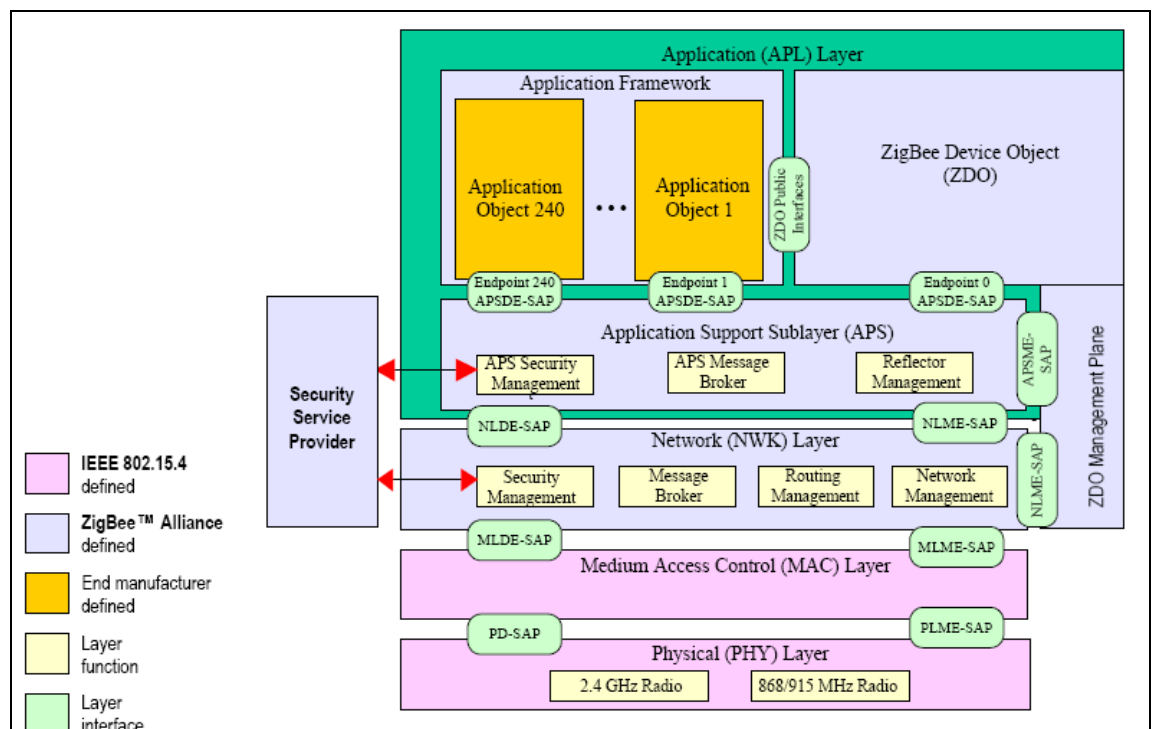
Kuvassa 19 on esitetty protokollan rakenne.



Kuva 19. ZigBee-protokollan rakenne [14]

Jokainen protokollapinon kerroksista tarjoaa palveluja yläpuolella olevalle pinon kerrokselle. Kerrosten välillä on liitynnät erityisten SAPien eli Service Access Pointtien kautta.

Kuvassa 20 on esitetty protokollapinon rakenne hieman tarkemmin. IEEE:n standardissa 802.15.4 fyysisen kerroksen (PHY) määrittelyssä on kaksi erillistä määrittelyä. 2,4 GHz:n taajuusalueella toimiva määrittely (250 kb/s) on maailmanlaajuinen ja toinen määrittely on kahdella taajuusalueella; 868 MHz:n taajuuskaista (20 kb/s) Eurooppaa varten sekä 915 MHz:n taajuuskaista (40 kb/s) Pohjois-Amerikkaa ja Australiaa varten.



Kuva 20. ZigBee-protokolla. [15, s. 30]

Taajuuskaistat on jaettu kanaviin siten, että Euroopan 868 MHz:n taajuuskaistalla on yksi kanava, Pohjois-Amerikan 915 MHz:n taajuuskaistalla 10 kanavaa ja maailmanlaajuisella 2,4 GHz:n taajuuskaistalla 16 kanavaa. Kaikki kuuluvat ns. ISM-taajuuksiin ja ovat siten lupavapaita pienillä lähetystehoilla. (868 MHz:n taajuusalueella $P < 25 \text{ mW}$ ja 2,4 GHz:n taajuusalueella $P < 10 \text{ mW}$ [16]).

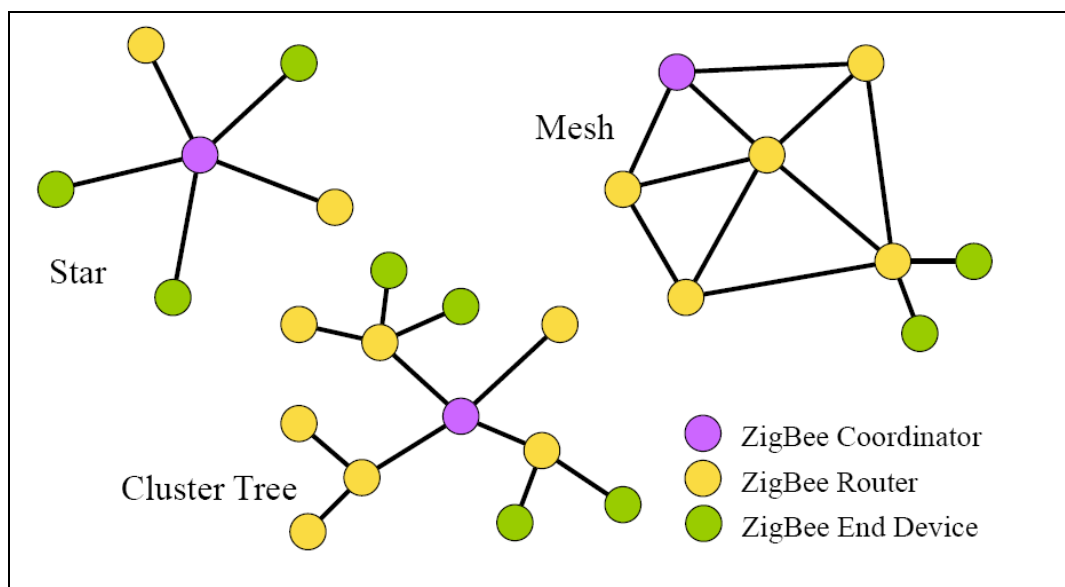
ZigBee-allianssin periaatteena on tarjota 802.15.4-standardin pohjalta rakennettu protokollastandardi, johon jokainen protokollaa käyttävä voi tehdä oman sovelluksensa (kuvassa 20. application object1...240), sovelluskerroksen toimiessa/antaessa kehukset langattomalle tiedonsiirrolle. [15.]

3.1.1 ZigBee-verkon rakenne

ZigBee-protokolla tukee 64-bittisiä verkko-osoitteita, eli olemassa olevia osoitteita voi olla 2^{64} kpl. Käytännössä yhdessä verkossa maksimimäärä on 2^{16} eli 65536 laitetta. Verkossa olevilla laitteilla voi olla erilaiset profiilit; laite voi olla FFD, eli Full Function Device, tai RDF, eli Reduced Function Device. FFD-laite omaa kaikki protokollan toiminnot ja voidaan edelleen asettaa toiminaan joko verkon koordinaattorina tai verkon reitittimenä. RDF-laitteen toimintoja on rajattu, ja tällainen laite toimii vain päätelaitteena. FFD- ja RDF-profilien laitteet voidaan asettaa toimimaan seuraavasti:

- 1) Laite voi olla verkon koordinaattori (Zigbee coordinator /ZC). Jokaisessa verkossa tulee olla yksi koordinaattori, joka on aloitteellinen laite verkon muodostuksessa. Koordinaattori sallii muiden laitteiden liittyä tietyin ehdoin verkkoon, sekä sallii datan reitittämisen itsensä kautta.
- 2.) Laite voi olla reititin (Zigbee router /ZR). Reititin sallii tietyin ehdoin muiden laitteiden muodostavan yhteyden sen kanssa, sekä sallii datan reitittämisen itsensä kautta.
- 3) Laite voi olla päätelaite (Zigbee End device /ZED). Päätelaite ei salli liittymistä eikä reititystä itsensä kautta. Laite voi ”keskustella” vain FFD-laitteen eli joko reitittimen tai koordinaattorin kanssa.

ZigBeen verkkokerros (NWK) tukee kuvassa 21 esitettyjä verkkorakenteita; 1) tähti-, 2) puu- ja 3) verkkotopologioita.



Kuva 21. ZigBeen verkkotopologiat [17]

Verkon liikennöinti tapahtuu CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance)-siirtotien varausmenetelmää käyttäen. CSMA-CA-menettelyssä lähettävä laite kuuntelee siirtotietä, ja jos siirtotie on vapaa, laite varaa siirtotien ja alkaa lähettää omaa dataa siirtotielle. Mikäli siirtotie on varattu, laite odottaa satunnaisen ajan ja kuuntelee sitten siirtotietä uudelleen.

ZigBeen tiedonsiirron salaus on kolmetasoinen. Tiedonsiirto laitteiden välillä voi olla kokonaan salaamatonta, pääsyylistoihin perustuva, tai tiedonsiirrossa voidaan käyttää 128-bittistä AES-salausta. [17]

3.1.2 ZigBeen käyttösovellukset

ZigBee-protokollan kehittäminen on lähtenyt tarpeesta luoda pienille tiedonsiirtonopeuksille lyhyen kantaman verkko, joka toimii hyvin vähällä virralla. Yksi käyttöalue ZigBeelle on erilaiset laitteet, jotka viettävät suuren osan toiminta-ajastaan ns. stand-by-tilassa. Sellaisia laitteita ovat esimerkiksi palovaroittimet, langattomat termostaatit, viihdelaitteiden kauko-ohjaimet tai vaikkapa valaistuksen ohjaus. Tällaisten laitteiden tulisi toimia AA-kokoisella

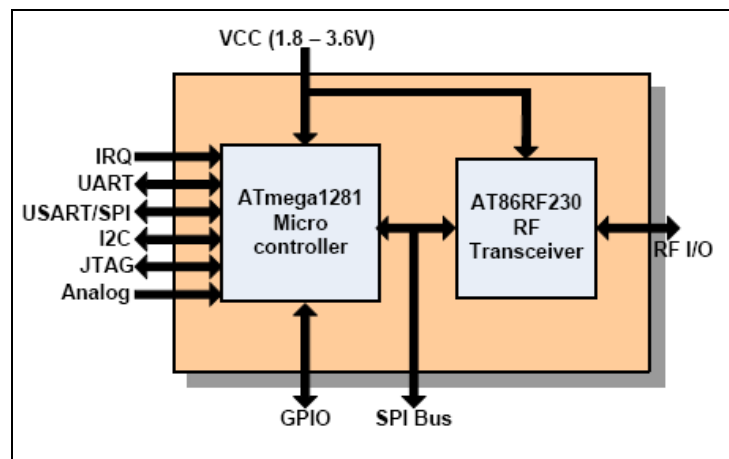
paristolla jopa vuosia. Toinen käyttöalue on teollisuuden mittaukset, joissa tarvitaan runsaasti antureita ja halutaan välttää tai ei voida käyttää anturiviestien välittämiseen kaapelointia. Göteborgin kaupungin energialaitos käyttää ZigBee-tekniikkaa asukkaiden sähkömittarien kaukoluennassa.

Lyhyen kantaman langattomien tekniikoiden tulevaisuus näyttää erittäin valoisalta. Ympäristöä tarkkaillaan erilaisilla sensoreilla ja antureilla yhä enemmän. Tunnettu futurologi Paul Saffo on sanonut: - Kuten PC oli 80-luvun symboli ja WWW-tekniikka 90-luvun symboli, tulee seuraava murros olemaan halpojen sensoreiden yleistymisen [17].

Sensorit pitää lukea joko langallisesti tai langattomasti, ja tähän saumaan ZigBee langattomana sekä edullisena ”standardina” on vartenotettava vaihtoehto.

3.2 Meshnetics ZigBit

Meshnetics on venäläinen ZigBee-moduuleja ja ohjelmistoja kehittävä yritys. Meshnetics on keskittynyt käyttämään sovelluksissaan Atmelin valmistamia piirejä, joista se on koonnut piirilevyille asennettavia valmiita moduuleja. Meshneticsin moduulit koostuvat Atmelin radiopiiristä ATRF230 sekä mikrokontrollerista ATmega1281 (kuva 22). Lisäksi joissakin malleissa on lisätty vahvistin. Meshnetics on nimennyt moduulinsa ZigBitiksi.



Kuva 22. Meshnetics ZDM-A1281-B0 ZigBee-moduulin rakenne [18]

Radiomoduuleja on saatavissa 2,4 GHz:n tai 868/915 MHz:n versiona. 2,4 GHz:n versiota on saatavilla ilman antennia, kahdella siruantennilla tai UFL-ulkoisen antennin liitynnällä varustettuna. 868/915 MHz:n versiota saa vain ilman antennia.

Ilman antennia varustetut moduulit on tarkoitettu asennettavaksi piirilevyille, johon on tehty piirilevyantenni.

2,4 GHz:n ulkoisella UFL-antenniliitännällä ja vahvistimella varustetulle moduulille luvataan hyvissä olosuhteissa jopa 4000 metrin kantama (Line of sight), mikä on varsin kova lukema, sillä alun perin ZigBee-laitteiden kantamaksi on arveltu riittävän 100 metriä.



Kuva 23. 2,4 GHz:n vahvistimella varustettu ZigBee-moduuli [19]

Meshnetics on tarjonnut ZigBee-moduuleihinsa omaa protokollapino-ohjelmistoa. Ensimmäinen versio oli nimeltään eZeeNet, ja toinen, kehittyneempi ja nyt käytössä oleva on nimeltään BitCloud.

Itse ZigBit-moduulit sisältävät ATmega 1281-ytimen, joka pyörittää protokollapinoa. Ulkoisia liityntöjä esimerkiksi tässä työssä käytetyssä ZDM-A1281-A2-moduulissa on seuraavasti:

- 9 kpl yleiskäyttöisiä I/O-pinnejä
- 2 kpl keskeytyslinjaa
- 3 linjaa AD-muunninta varten
- 1-wire-liitäntä
- yksi UART- ja yksi USART-liitäntä
- IIC-väylä sekä

- JTAG-liitäntä moduulin ohjelmointia varten

Meshnetics tarjoaa kehitykseen kuvan 24 mukaista kehityspakettia, jossa on kolme ZigBit-kehityskorttia sekä ohjelmisto-CD. Ohjelmisto-CD:llä on BitCloud-protokollapino suoraan ladattavana .hex-koodina sekä lähdekoodit niistä ohjelmiston osista, joihin on arveltu asiakkaan tekävän muutoksia. Ihan kaikesta ohjelmiston osista ei lähdekoodeja kuitenkaan löydy. Kehityskorttien testaamista varten paketissa on WSN Monitor -ohjelma, joka PC:lle asennettuna mahdollistaa verkon tutkimisen, sekä sillä voidaan tarkastella kehityskorttien lämpötila- ja valaistusanturien arvoja. Kehityspaketin ostajalle luvataan myös on-line-tukea, jolloin kysymyksiin luvataan vastata sähköpostin välityksellä.



Kuva 24. ZigBit-kehityspaketti [19]

Vuoden 2009 helmikuussa julkistettiin tieto, jonka mukaan Atmel on ostanut Meshneticsin IP-oikeudet ja jatkaa ZigBit-moduulien ja BitCloud-ohjelmiston kehittämistä. Nähtäväksi jää, miten tuo kauppa vaikuttaa Meshneticsiin.

4 LANGATTOMAN TÄRINÄMITTARIN MITTAUSVASTAANOTIN

4.1 Suunnittelu

Työn suunnitteluvaiheen alusta lähtien oli selvää, ettei rajallinen kehitysaika mahdollistaisi koko järjestelmän (=mittausvastaanotin ja anturit) kehittämistä. Tällöin työ rajattiin siten, että insinöörityönä rakentaisiin järjestelmään mittausvastaanottimen. Tämä siksi, että näin oli mahdollista saada työksi kohtalaisen monipuolinen työ, sisältäen lähes kaikki prototyyppilaitteen rakentamisen vaiheet; vaatimusten määrittely, laitteiston ja ohjelmiston suunnittelu sekä laitteen prototyypin rakentaminen ja ohjelmointi.

4.1.1 Vaatimusten määrittely

Suunnittelun alussa määritettiin karkealla tasolla vaatimukset, jotka laitteen tuli täyttää. Näitä vaatimuksia tarkennettiin suunnittelun aikana. Aluksi mittausvastaanottimelle määritettiin seuraavat vaatimukset:

- 1) Mittausvastaanottimen tulee toimia käyttöliittymänä tärinäantureista koostuvaan verkkoon, ja sen kautta tulee voida:
 - a. Lisätä ja poistaa antureita verkkoon/verkosta.
 - b. Pitää listaa verkossa olevista antureista.
 - c. Vastata oikean kellonajan jakamisesta antureille.
 - d. Lukea ja poistaa anturien tallentamia mittaustuloksia.
- 2) Mittausvastaanottimessa tulee olla riittävän suuri näyttö, jotta:
 - a. Oikea päiväys ja kellonaika sekä mittauksen asetukset ovat yhdellä kertaa nähtävillä mittauksen aikana.
 - b. Vastaanottimen kaikki valikkonäkymät sopivat yhdellä kertaa laitteen näyttöön.

- 3) Mittausvastaanottimen näppäimistön tulee olla riittävän yksinkertainen, mutta kuitenkin käyttäjäystävällinen:
 - a. Yksi toiminto tapahtuu yhdestä näppäimestä.
 - b. Näppäimistö on looginen.
 - c. Näppäimet ovat riittävän isoja.
- 4) Vastaanottimessa (ja antureissa) tulee olla reaaliaikakello, jotta mittaustapahtumat voidaan varustaa aikaleimalla ja vastaanottimessa tulee olla mahdollisuus tehdä kellon ja päivämäärän asetuksia.
- 5) Vastaanottimessa tulee olla mahdollisuus liittää se PC-tietokoneeseen
- 6) Vastaanottimessa tulee olla mahdollisuus ”säätää” mittaustapahtuman parametreja, kuten mittauksen kestoa ja liipaisuarvoa.

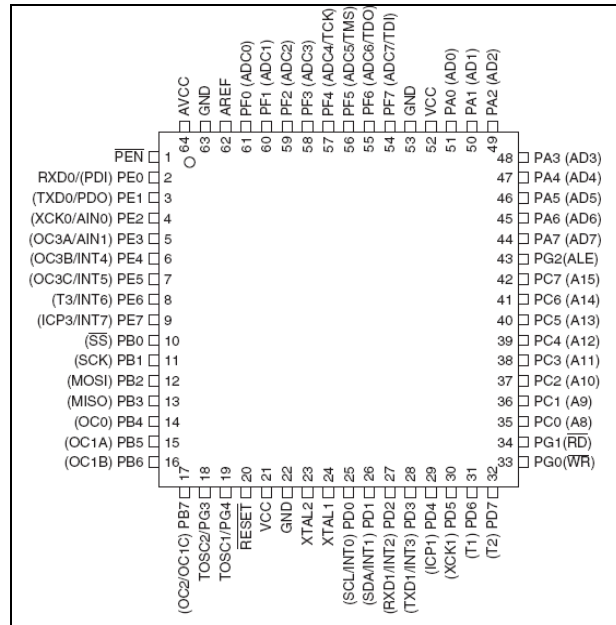
Nuo vaatimukset toimivat pohjana, jonka mukaan laitteesta tehtiin lohkokaavio ja sen myötä ryhdyttiin etsimään sopivia komponentteja laitteeseen.

4.1.2 Laitteen suunnittelu

Laitteistosuunnittelun lähtökohdaksi otettiin Meshneticsin valmistama ZigBee-moduuli, koska moduulin sisällä olevan ATmega 1281 -mikrokontrollerin rakenne ja ominaisuudet olivat tulleet tutuiksi monissa aikaisemmissa laboratoriotöissä.

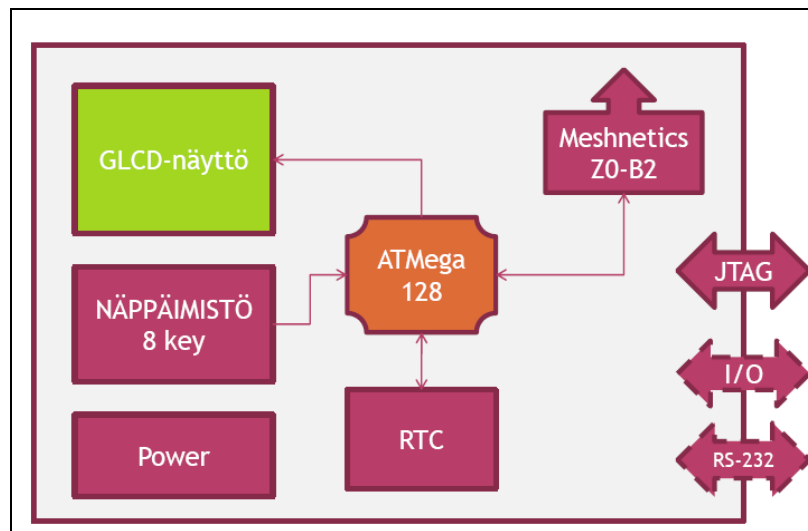
Aikaisempien kokemusten ja vaatimusten määrittelyn perusteella todettiin, että pelkkä ZDM-A1281-B2 ZigBee-moduuli ei riittäisi ohjaamaan kaikkea laitteen toiminnallisuutta. Näin ollen laitteessa tuli olla toinen mikrokontrolleri, joka ohjaa kaikkea laitteen toiminnallisuutta, jolloin ZigBee-moduulin tehtäväksi jäisi ainoastaan tiedonsiirto anturien ja päätelaitteen välillä.

Laitteen ”pääohjaimeksi” valittiin Atmelin ATmega 128 -mikrokontrolleri (kuva 25), jolloin oli mahdollista käyttää samoja työkaluja molempien, sekä pääohjaimen että ZigBee-moduulin, ohjelmointiin ja debuggaukseen.



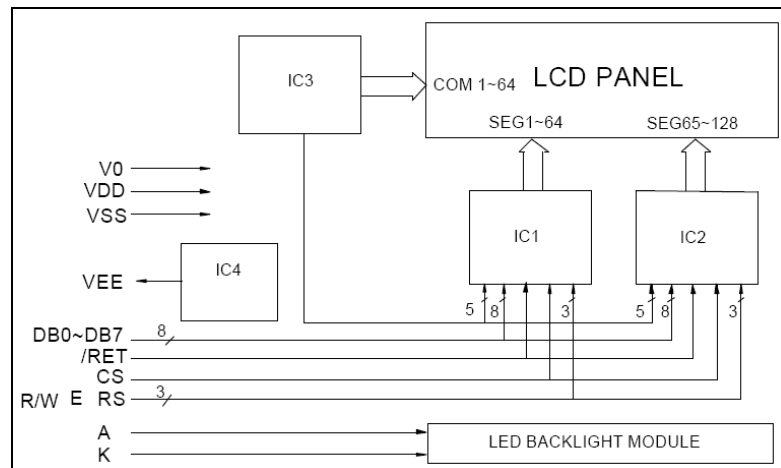
Kuva 25. ATmega128, pinnijärjestys [20]

Kun laitteen kaksi pääkomponenttia oli määrätty, piirrettiin laitteesta lohkokaavio (kuva 26)



Kuva 26. Mittausvastaanottimen lohkokaavio

Näyttönä laitteessa toimisi graafinen 128 x 64-kokoinen näyttö, jollainen hankittiin edullisesti yhdysvaltalaisesta Sparkfun-nettikaupasta (www.sparkfun.com). Näyttö on kiinalaisvalmisteen GDM12864H, johon mahtuu vaikkapa kahdeksan 8 pisteen korkuista merkkiriviä. Näytön ohjauksesta vastaa kaksi KSO0108-näytönohjauspiiriä ja näyttö on jaettu keskeltä, jolloin molemmat ohjainpiirit ohjaavat 64 x 64-kokoista näyttöaluetta (kuva 27)



Kuva 27. GDM12864H-näytön lohkokaavio [21]

Tässä näytössä on huomioitavaa, että se on ”piste pisteeltä” ohjattava eikä esimerkiksi sisällä ROM-muistia, jossa olisi valmiita merkkejä muistissa. Näin ollen näytöllä näytettävät merkit tulee olla näyttöä ohjaavan prosessorin muistissa. Hyvä puoli näytössä on, että näytöllä näytettäviä tekstejä, kuvia ja kuvioita ei ole mitenkään rajattu (muutoin kuin kokonsa puolesta) ja Internetistä voi löytää useita PC:lle asennettavia LCD-editoreja, joilla voi loihtia vaikkapa traktorin näyttöön, kuten kuvassa 28.



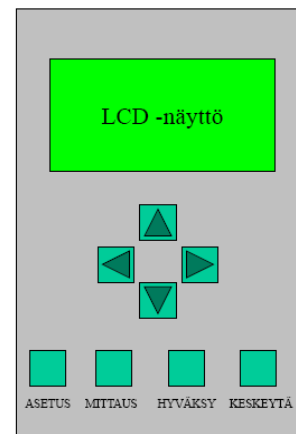
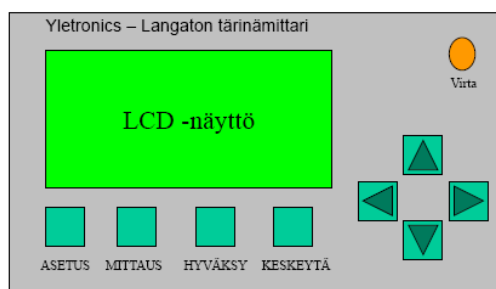
Kuva 28. GDM12864H 128 x 64 graafinen LCD-näyttö [25]

Näppäimistön suunnittelussa lähdettiin liikkeelle ajatuksesta tehdä näppäimistö, jossa yhdestä näppäimestä ohjattaisiin yhtä toimintoa. Samoin tuossa vaiheessa oli jo hahmoteltu laitteen valikkorakennetta ja määritelty laitteeseen kaksi ”päävalikkoa”: Asetusvalikon sekä Mittausvalikon. Näistä lähtökohdista päätettiin tehdä 8-näppäinen näppäimistö, jossa olisi seuraavat näppäimet:

- 1) ENTER (Hyväksy) toimintojen hyväksymiseen/vahvistamiseen,
- 2) BACK (takaisin) valikoissa takaisinpäin siirtymiseen
- 3) Nuolinäppäimet: Ylös, Alas, Vasemmalle ja Oikealle valikoiden siirtymiseen ja niissä olevien arvojen muuttamista varten.

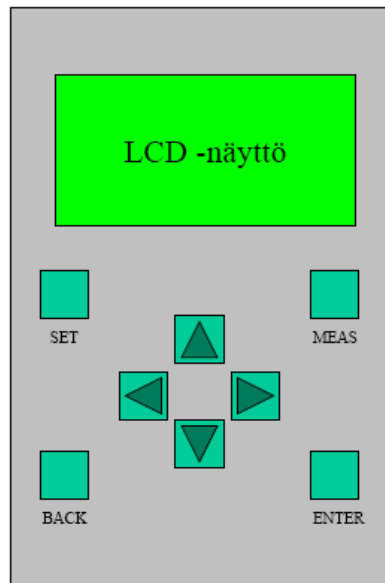
Lisäksi molemmille valikoille haluttiin omat näppäimet, jolloin painamalla MEAS-näppäintä pääsisi suoraan mittausvalikkoon, sekä SET-näppäimellä asetusvalikkoon.

Laitteen ulkonäköä ja näppäimien sijoittelua hahmoteltiin Microsoftin PowerPoint –ohjelmalla, ja kuvassa 29 on kaksi suunnittelun ”väliversiota”.



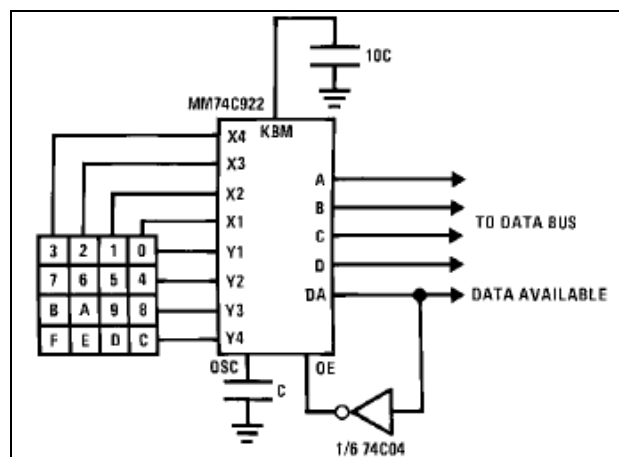
Kuva 29. Laitteen käyttöliittymän eri suunnitteluversioita

Lopullisesti näyttö-konfiguraatio hahmottui, kun laitetta varten saatiin 175 x 120 x 50 mm-kokoinen laitekotelo. Laitteen ulkonäkö tulisi olemaan suurin piirtein kuvan 30 mukainen.



Kuva 30. Laitteen käyttöliittymän suunnittelun lopputulos

Itse näppäimistön kytkentään valittiin näppäimistökooderi MM74C922. Kytkeä tehtiin kuvan 31 mukaiseksi sillä erotuksella, että näppäimiä päätelaitteessa olisi vain kahdeksan.

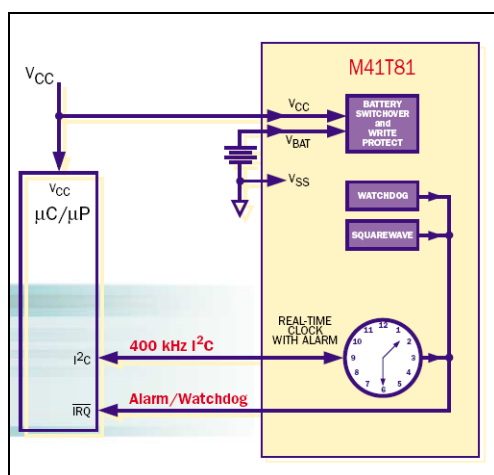


Kuva 31. Näppäimistökooderin kytkentä [22]

Näppäimistö on asynkroninen, eli näppäimen painallus aiheuttaa DA-linjaan 1-tilan, joka vuorostaan aiheuttaa mikrokontrollerin keskeytystulossa keskeytyksen. Tämän jälkeen kes-

keytysaliohjelma käy lukemassa näppäimistöön kytketyn portin tilan, jolloin ohjelma saa selville, mitä näppäintä on painettu. Liitteessä 2 on näppäimistön täydellinen kytkentäkaavio.

Jotta mittausjärjestelmän mittaamat värinäarvot saadaan luotettavasti yksilöityä, tulee järjestelmässä olla kello, jonka avulla tallennettuihin mittaustiedostoihin saadaan aikaleimat. Kellopiiriksi tähän laitteeseen valittiin ST Microelectronicsin valmistama M41T81-reaaliaikakellopiiri, jollainen löytyi suoraan tietotekniikan laboratorion komponenttivarastosta. M41T81-kellopiiri liitetään prosessorin IIC-väylään, ja lisäksi piiri tarvitsee oman 32 768 Hz:n kiteen sekä 3 voltin muistinvarmennuspariston. Kuvassa 31 on esitetty RTC-kellopiirin kytkentä.



Kuva 31. M41T81-kellopiirin kytkentä [23]

Tiedonsiirto laitteen pääkontrollerin, ATmega128:n ja ZigBee-moduulin välillä tuli toimia UART-väylää hyväksikäyttäen. Tässä oli kuitenkin ongelmana sovittaa piirien jännitetasot yhteen. ATmega128:lla tuli käyttää +5 voltin jännitettä LCD-näytön takia, ja ZigBee-moduulin käyttöjännite olisi +3 voltia. Jotta jännitteet saataisiin molemmille laitteille sopiviksi, suunniteltiin väliin tasonmuuntimet käyttäen hyväksi puskuripiiriä 74LX1G07. Puskuripiirejä tulisi väylään kaksi (TX- ja RX-linjat).

ZigBee-moduuli tarvitsee itselleen verkko-osoitteen ja sen muodostamiseksi oman ID-tunnuksen. ID-tunnus saadaan moduuliin kytkemällä sen 1-wire-väylään DS2411 ID-piiri. DS2411 on Dallas Semiconductorin valmistama piiri, jossa kullekin piirille on valmistusvaiheessa ohjelmoitu yksilöllinen 64-bittinen rekisteröintiosoite, jota ei voida muuttaa miten-

kään, esimerkiksi uudelleen ohjelmoimalla. Dallas takaa, ettei kahta samannumeroista piiriä ole olemassa.

4.1.3 Kytkenän ja piirilevyn suunnittelu

Kun komponentit oli valittu, oli vuorossa laitteen kytkentäkaavion piirtäminen sekä piirilevyn suunnittelu ja valmistus.

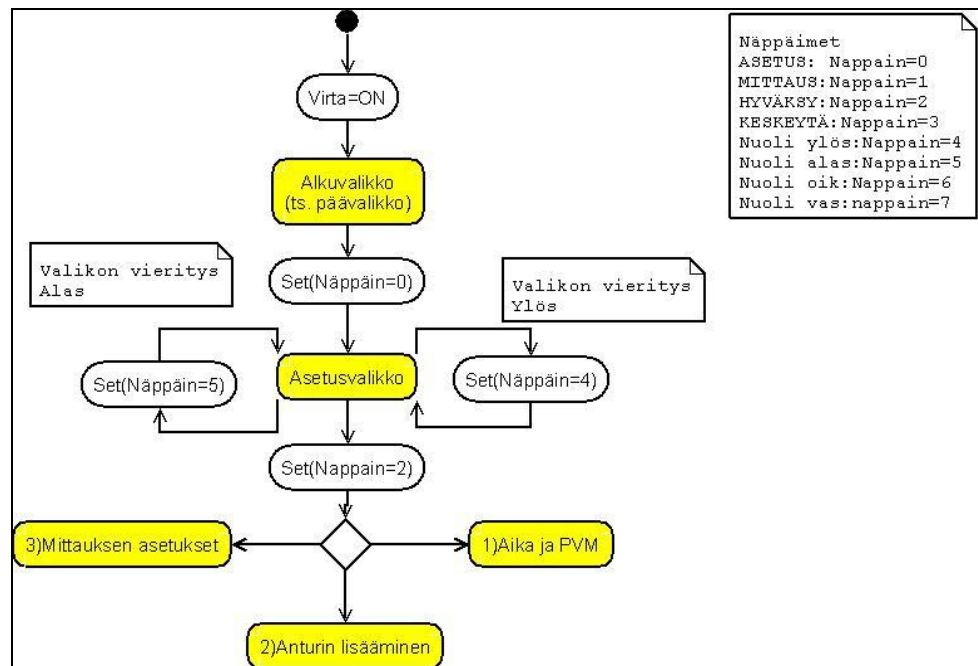
Koko piirilevyn suunnitteluprosessi, kytkentäkaaviosta piirilevyn suunnitteluun, tehtiin PADS 2005 -ohjelmistolla. Kytkenäkaavio suunniteltiin PADS logic -ohjelmalla, josta komponenttien loogiset kuvaukset ja niiden kytkennät siirrettiin PADS layout -ohjelmaan, jossa tapahtui komponenttien paikalleen sijoittaminen. Lopuksi komponenttien väliset johdotukset tehtiin PADS router -ohjelmalla.

Piirilevyn suunnittelussa olivat suunnittelua rajaavina tekijöinä piirilevyn koko sekä kerrosten määrä. Piirilevyn tuli mahtua valitun laitekotelon sisään. Lisäksi piirilevyn tuli olla korkeintaan 2-puolinen, jotta se oli mahdollista valmistaa koulun tuotantotekniikan laboratorion välinein.

Laitteen varsinaisen piirilevyn kytkentäkaaviot ovat liitteessä 1 ja näppäimistön liitteessä 2.

4.1.4 Ohjelmiston suunnittelu

Laitteen ohjelmistoa suunniteltiin jo erään ohjelmointikurssin ohessa, ja tuolloin saatiin suunniteltua laitteen valikot. Laitteen ohjaus suunniteltiin toteutettavaksi tilakone-periaatteella, jossa laite on aina jossain tilassa ja odottaa käyttäjältä tai laitteistolta herätettä, jolla siirrytään seuraavaan tilaan. Esimerkiksi kuvassa 32 on esitetty ohjelman toiminta käynnistettäessä laite ja siirryttäessä asetusvalikkoon sekä valikon ”rullaaminen”.



Kuva 32. Käynnistys ja asetusvalikkoon siirtyminen

Liitteessä 3 on esitetty tilakaaviot tärkeimmistä toiminnoista.

4.2 Laitteen rakentaminen

Laitteen piirilevyn ensimmäinen versio yritettiin tehdä syövyttämismenetelmällä. Koska piirilevyn johtimien välit olivat paikoin niin ohuet ettei, levy syöpynyt kaikista väleistä, päätettiin levy lopulta jyrsiä koulun Bungard-merkkisellä piirilevyjyrsimellä.

Laitteen kokoaminen alkoi piirilevyn sähköisellä testaamisella katkosten ja oikosulkujen varalta. Testaaminen tehtiin suurentavan lampun alla yleismittarin avulla.

Piirilevyn testauksen jälkeen juotettiin levyille 3 V:n regulaattori, jonka jälkeen levyille kytkettiin 5 voltin käyttöjännite ja testattiin yleismittarilla käyttöjännitteet piirien käyttöjännitetasojen pädeiltä.

Kun piirilevyn toiminta ja käyttöjännitteet oli testattu, oli vuorossa piirilevyn kalustaminen. Kalustaminen aloitettiin juottamalla pienimmät mikropiirit paikoilleen mikroskooppia hyväksikäyttäen. Pienijalkaiset komponentit juotettiin ensimmäiseksi, jotta isommat komponentit eivät olisi tarkan työskentelyn tiellä. Lopuksi juotettiin levyille vastukset ja kondensaattorit sekä viimeiseksi liittimet.

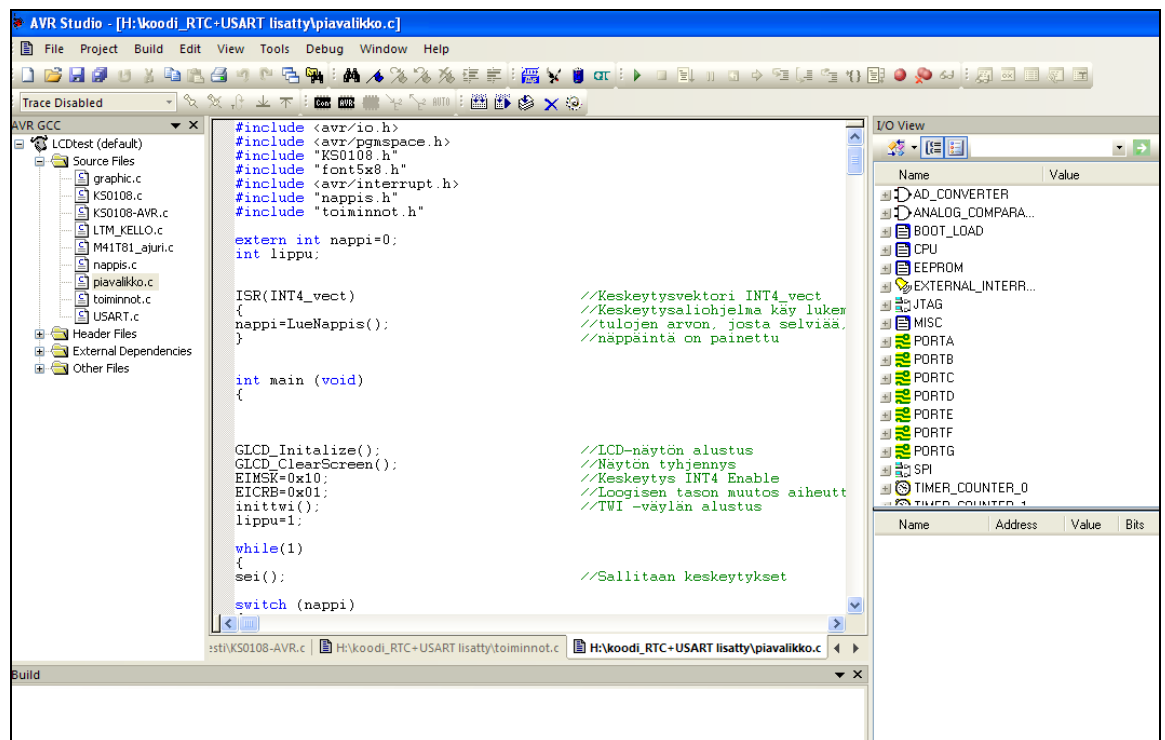
Kun piirilevy oli kalustettu, tehtiin näyttöä varten 20-johtiminen lattaakaapeli ja kiinnitettiin se piirilevylle. Toimintoja ohjaava ATmega128 oli jo ollut käytössä ja sitä oli ohjelmoitu STK500 kehitysalustassa. Mm. näytön ajurit ja osa valikoista oli jo ohjelmoitu valmiiksi. Kun virta kytkettiin piirilevylle, todettiin, että näytölle ilmestyi laitteen ”perusvalikko”.

Tämän jälkeen suunniteltiin piirilevy näppäimistöä varten ja jysrittiin se Bungard-piirilevyjysrimellä. Näppäimistön kokoaminen oli helppoa, koska siinä ei ollut pieniä, mikroskoopin alla juotettavia osia. Näppäimistöpiirilevy liitettiin varsinaiseen laitteen piirilevyyn nelinapaisen datakaapelin avulla. Myös näppäimistön virransyöttö tapahtuu kaksinapaisen virtakaapelin avulla varsinaisen laitteen piirilevyltä.

Laitteen koteloa muokattiin leikkaamalla kotelon kanteen aukko LCD-näyttöä varten sekä poraamalla reiät näppäinten kohdalle. Porauksen kohdistuksessa käytettiin mallina näppäimistön piirilevyä, joista tarkasti mittaamalla saatiin yksittäisten näppäinten paikat. Näppäimistön kalvo tehtiin Microsoftin PowerPoint -näyttöesitysohjelmalla ja tulostettiin värityslostimella paperille, joka vielä suojattiin muovikalvolla. Tässä protoversiossa kalvon näppäinten paikat eivät osuneet aivan ”millilleen” kohdilleen, mutta käytännön testaukseen tarkkuus oli riittävä. Liitteen 4 kuvissa on esitetty ”valmis” laite, laitteen koteloointi sekä piirilevyt.

4.3 Laitteen ohjelmointi ja testaus

Laitteen molempien ohjaimien, ATmega128:n ja ZigBee-moduulin, ohjelmointi tehtiin kuvassa 33 olevalla Atmelin AVR Studio 4 -ohjelmointiympäristöllä, johon on lisäkomponenttina asennettu WinAVR-kääntäjäohjelma. Ohjelmointikielenä käytettiin C-kieltä.



Kuva 33. AVR Studio 4 -ohjelmointiympäristö

Molempien ohjainten ohjelman lataaminen tehtiin Atmelin JTAGICE mk2 -ohjelmointilaitteella, käyttäen hyväksi ohjainten JTAG-liitäntää. ATmega128-kontrolleri voidaan kyllä ohjelmoida muullakin tavoin, mutta ZigBee-moduulin ohjelmointi onnistuu vain JTAG-liitäntän kautta.

Aikataulullisista syistä pyrittiin löytämään ja käyttämään niin paljon valmista ohjelmakoodia kuin mahdollista. ATmega128-kontrolleriin löytyi graafisen LCD-näytön ohjauskoodi sekä siihen liittyvät .h-tiedostot ja merkkien määrittelyt puolalaisen Radoslaw Kwiecienin kotisivuilta[24], joka on antanut koodinsa vapaaseen levitykseen. Radoslawin koodia piti muokata hieman esimerkiksi ATmega128:n ohjauspinnien osalta.

Valikoiden ohjelmoinnissa oli hankaluutena rakentaa toimiva koodi valikoiden vieritykseen sekä valikoiden sisällä olevien arvojen muuttamiseen. Ratkaisu löytyi käyttämällä sisäkkäisiä

switch-case- ja if-rakenteita. Liitteessä 5 on esimerkkinä listaus ohjelmasta, jolla päivitetään päivämäärä, vuosi ja kellonaika. ZigBit-moduulin ohjelmoinnissa muokattiin Meshneticsin valmiita ohjelmakoodia, jotta saatiin mm. määritettyä laitteelle eri rooleja (Koordinaattori/reititin) verkon toiminnan testausta varten.

Koska ohjelmakoodia tähän laitteeseen syntyi A4-kokoon koostettuna useita kymmeniä sivuja, ei liene tarpeellista liittää kaikkea ohjelmakoodia edes liitteeksi. ZigBit-moduulin ohjelmakoodit voi jokainen ladata ilmaiseksi Atmelin tai Meshneticsin kotisivuilta.

5 TYÖN TAVOITTEIDEN SAAVUTTAMINEN JA LANGATTOMAN TÄRINÄMITTARIN JATKOKEHITYS

5.1 Työn tulokset

Langaton tärinämittari koostuu useasta laitteesta. Tässä insinööriyössä kehitettiin vasta mitausvastaanottimen ensimmäinen prototyyppi. Jotta tärinöitä langattomasti mittaava järjestelmä saadaan toimivaksi, tulee siihen liittyvä anturi myös kehittää toimivaksi laitteeksi.

Työssä ei ehditty kehittämään aivan kaikkia suunniteltuja toiminnallisuuksia, eikä aika riittänyt myöskään esimerkiksi oikeaan testaamiseen. Käytännössä laitteen testaus oli laitteen toiminnan kokeilua kehitysvaiheiden välillä. Eniten työssä harmaita hiuksia aiheutti Meshneticsin protokollapino-ohjelmisto ja kunnollisen dokumentaation puuttuminen. Erityisesti olisi kaivattu enemmän tietoa siitä, kuinka moduulia voi ohjata UART-liitäntän kautta toisella mikrokontrollerilla. Meshneticsin ZigBee-moduuli ja protokolla-ohjelma BitCloud toimivat kuitenkin PC-testiohjelman WSN-monitorin kanssa moitteettomasti, verkkoon liittyminen onnistui nopeasti ja kokeiluissa yhteys toimi laboratorion sisällä päätelaitteen ja ZigBit-kokeilukorttien välillä katkeamatta.

Työ oli omasta mielestäni vaativin tähänastisista töistä insinööritieteiden saralla. Erityisesti mikrokontrollerien ohjelmointi C-kielellä tuotti aluksi eniten harmaita hiuksia, mutta tässä myös kehitystä tapahtui eniten. Laitteen suunnittelu ja rakentaminen oli kohtalaisen suoraviivainen tehtävä, johtuen aikaisemmasta kokemuksesta vastaavien, tosin pienempien, prototyyppien rakentamisessa.

5.2 Jatkokehitys

Luonnollinen jatko laitteen kehittämisessä olisi suunnitella ja rakentaa prototyyppi langattomasta tärinäanturista. Tärinäanturin tulisi mielestäni täyttää seuraavat vaatimukset:

- Anturi olisi kooltaan pieni (maksimissaan ihmisen nyrkin kokoinen).

- Anturin hinta olisi loppukäyttäjälle 500–1000 euroa/anturi.
- Anturin tulee olla helppo asentaa ja käyttää.
- Anturin kotelon on oltava veden ja iskun kestävä.
- Anturin tulee toimia pienellä AA-paristolla tai akulla viikkoja.
- Anturin tulee tallentaa mittausdata esimerkiksi muistikortille ja erikseen pyydettyäessä lähettää tiedot päätelaitteelle.
- Anturissa tulee olla oma kello aikaleiman tallentamiseksi mittausarvojen ”kehystietoihin”.

Langattomia geofoneja on kehitelty, mutta jostain syystä ne eivät ole vielä ilmestyneet käyttöön, eivätkä myöskään rakennustärinöiden mittaukseen. Luulen, että ”avain” toimivaan langattomaan tärinänmittausjärjestelmään on siinä, että anturit eivät lähetä/yrityä lähettää mittausdataa reaaliajassa, vaan toimivat dataloggereina. Kun mittautapahtuma on ohi, anturit lähettävät mittaustiedot hallitusti päätelaitteelle tai tallentavat tiedot esimerkiksi muistikortille ja lähettävät ne päätelaitteelle erikseen pyydettyäessä.

Päätelaitteeseen tulisi kehittää liityntä PC:lle sekä määritellä ja kehittää järkevä tulosten esitystapa LCD-näytölle. Lisäksi kokemukset Meshneticsin ohjelmistosta, tai lähinnä vähäisestä dokumentaatiosta, ”rohkaisevat” perehtymään muiden valmistajien tekniikoihin. Esimerkiksi Texas Instrumentin komponentit sekä ohjelmistot vaikuttaisivat tutkimisen arvoisilta vaihtoehtoilta.

6 YHTEENVETO

Tässä työssä perehdyttiin tärinään fysikaalisena ilmiönä, aaltoliikkeenä. Maaperässä aaltoliikkeenä etenevä tärinä etenee väliaineesta riippuen pitkittäisenä, poikittaisena tai sekamuotoisena aaltoliikkeenä. Maaperässä etenevä tärinä voi olla aaltoliikkeen muodosta ja nopeudesta riippuen P-, S-, Rayleigh- tai Love-aaltoa.

Tärinöiden mittaukseen käytetään nykypäivänä monikanavaisia mittareita, joissa analoginen tärinäsignaali muutetaan digitaalseksi dataksi tallennusta varten. Rakennustärinöiden mittauksessa on vakiintunut käyttöön heilahdusnopeuden mittausta, joka korreloi parhaiten syntymien vaurioiden suuruuden kanssa.

Langaton ZigBee-anturiverkkoprotokolla perustuu IEEE 802.15.4-standardissa määriteltyjen PHY- ja MAC-kerrosten päälle rakennettuihin verkko- ja sovelluskerroksiin. Maailmanlaajuisesti 2,4 GHz:n taajuudella toimiva ZigBee-verkko tarjoaa hyvin energiatehokkaan tiedonsiirtomenetelmän, jolla voidaan siirtää dataa ja käskyjä erilaisten toimilaitteiden tai antureiden välillä noin 100 metriin asti jopa 250 kb/s nopeudella.

Insinööriyön käytännön osuudessa rakennettiin päätelaite tärinöitä mittaavaan ZigBee-anturiverkkoon. Kehitystyössä todettiin venäläisen Meshneticsin valmistama Bitcloud-protokollapino-ohjelma niukasti dokumentoiduksi. Käytössä olleet ZigBit-moduulit sekä BitCloud-ohjelmisto toimivat kuitenkin luotettavasti ja hyvin. Lisäksi rakennettu päätelaite saatiin liitettyä langattomaan anturiverkkoon ja verkon toiminta testattiin WSN-monitor -testiohjelmalla.

LÄHTEET

1. Wikipedia, Harmoninen värähtelijä [www -dokumentti]
http://fi.wikipedia.org/wiki/Harmoninen_v%C3%A4r%C3%A4htelij%C3%A4
2. Raimo Vuolio, Räjätystyöt 1991, 3. painos, ISBN 952-90-2761-3, SML ry kustantama.
3. Suomen kansallinen geologian komitea. [interaktiivinen www-dokumentti]
http://www.geologia.fi/index.php?option=com_content&task=view&id=97&Itemid=91
4. Science learning hub. Uuden-Seelannin tutkimusministeriön opetus projektin Internet-sivut [WWW-dokumentti]
http://www.sciencelearn.org.nz/contexts/earthquakes/sci_media/images/earth_waves
5. Professori Lauri J. Pesonen, Helsingin yliopiston fysiikan laitos. Luentomateriaali Geofysiikan johdantokurssiin 53501, 2008.
6. Helsingin yliopiston seismologian laitos. Tietoa maanjäristyksistä [www-dokumentti]
<http://www.seismo.helsinki.fi/fi/maanjtietoa/vastauksia.html>
7. Performance specifications for blasting seismographs, International society of explosive engineers/ Blast vibration and seismograph section, 2000, [www.isee.org]
8. Tampereen teknillinen yliopisto, Rakentamisen aiheuttamat värinät, lausuntoversio 3.8.2009. Liite 2. Värinämittareiden toiminnalliset vaatimukset ja tarkastusmenettely.
9. Hanna Leppänen, Diplomityö: Paalutuksesta ja pontituksesta aiheutuva värinä, 2008, Tampereen teknillinen yliopisto
10. Värinänmittausraportti 13.8.2009, Matti Hakulinen, Tampereen teknillinen yliopisto
11. Kuva: Stanford university, microstructures and sensors lab
12. Instandel Minimate Pro6, series IV, specifications sheet. 2009
13. Puhelinkeskustelu 26.10.2009 Mikko Yletyinen – Juha Tuovinen, Kalliotekniikka oy
14. ZigBee overview, ZigBee Alliance, 2009 [pdf -dokumentti]
15. ZigBee specification, ZigBee alliance, asiakirja nro. 05347r17/17. tammikuuta 2008
16. Määräys luvasta vapaiden radiolähettimien yhteistajuuksista ja käytöstä. Viestintävirasto 15Y/2008
17. ZigBee tutorial-wireless control that simply works, ZigBee alliance, joulukuu 2006 [PPT-dokumentti]
18. ZigBit ZDM-A1281-A2/B0 datasheet, Meshnetics

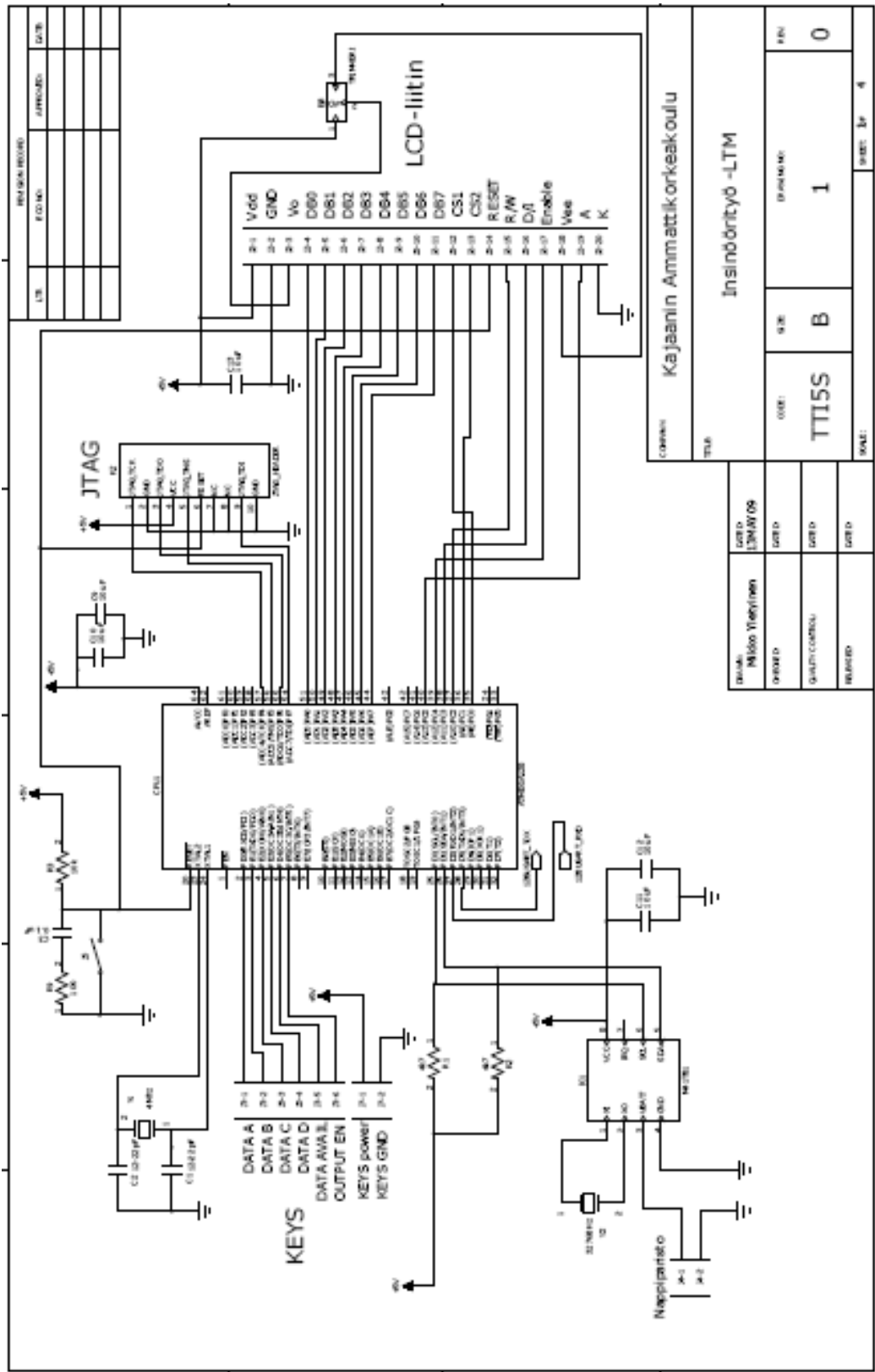
19. Meshnetics kotisivut, [www-dokumentti] www.meshnetics.com
20. ATMega128 datasheet, Atmel corporation 2008
21. GDM12864H datasheet, Xiamen Ocular, Ver A, 2006.12.11
22. MM74C922 datasheet, Fairchild Semiconductor, October 1987, revised April 2001
23. M41T81 datasheet, ST microelectronics, July 2002
24. Yleiskäyttöinen ohjelmakirjasto KS0108 graafisen LCD-näytön ohjaimelle, Radoslaw Kwiecienin kotisivut, 2009 [www-asiakirja] <http://en.radzio.dxp.pl/ks0108/index.html>
25. Kuvalähde tekijän omasta sähköisestä arkistosta, Alkuperä tuntematon

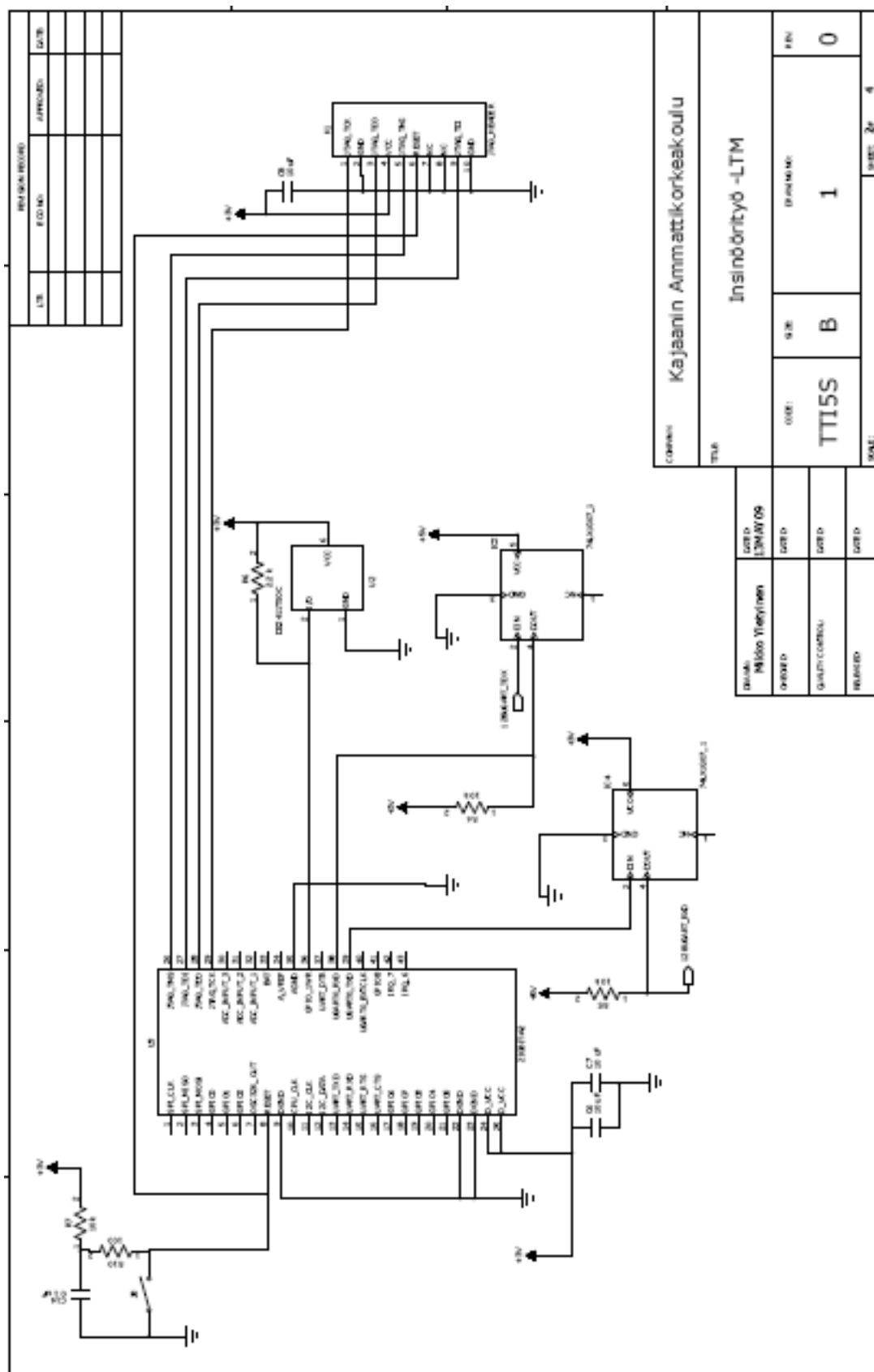


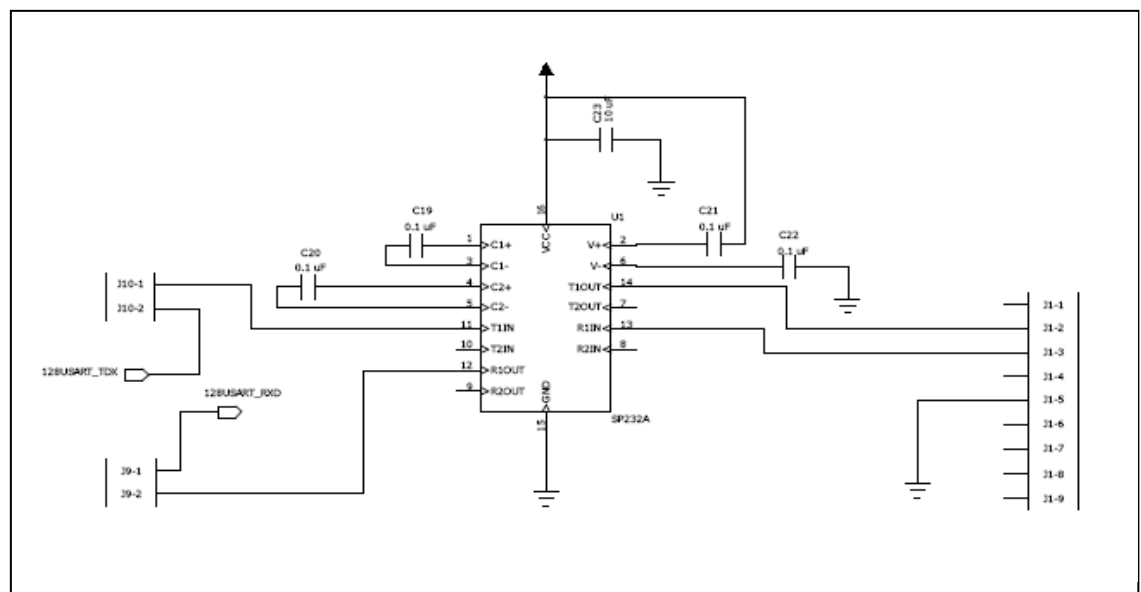
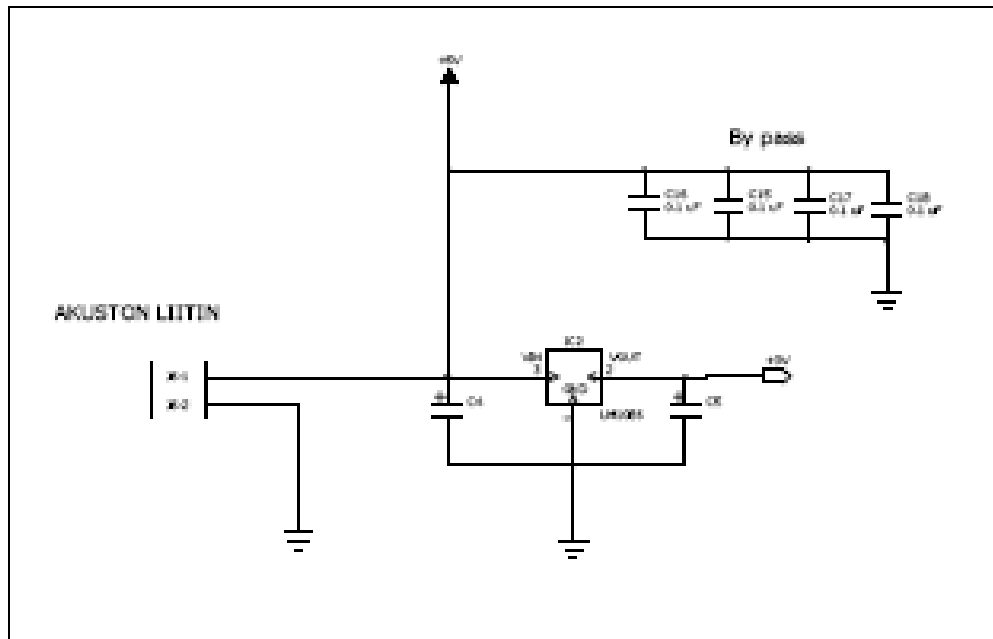
LIITTEIDEN LUETTELO

| | |
|---------|--|
| LIITE 1 | Päätelaitteen piirikaavio |
| LIITE 2 | Päätelaitteen näppäimistön piirikaavio |
| LIITE 3 | Ohjelmiston tärkeimmät tilakaaviot |
| LIITE 4 | Kuvat valmiista laitteesta |
| LIITE 5 | Esimerkkikoodi valikkorakenteesta |

PÄÄTELAITTEEN PIIRIKAAVIO

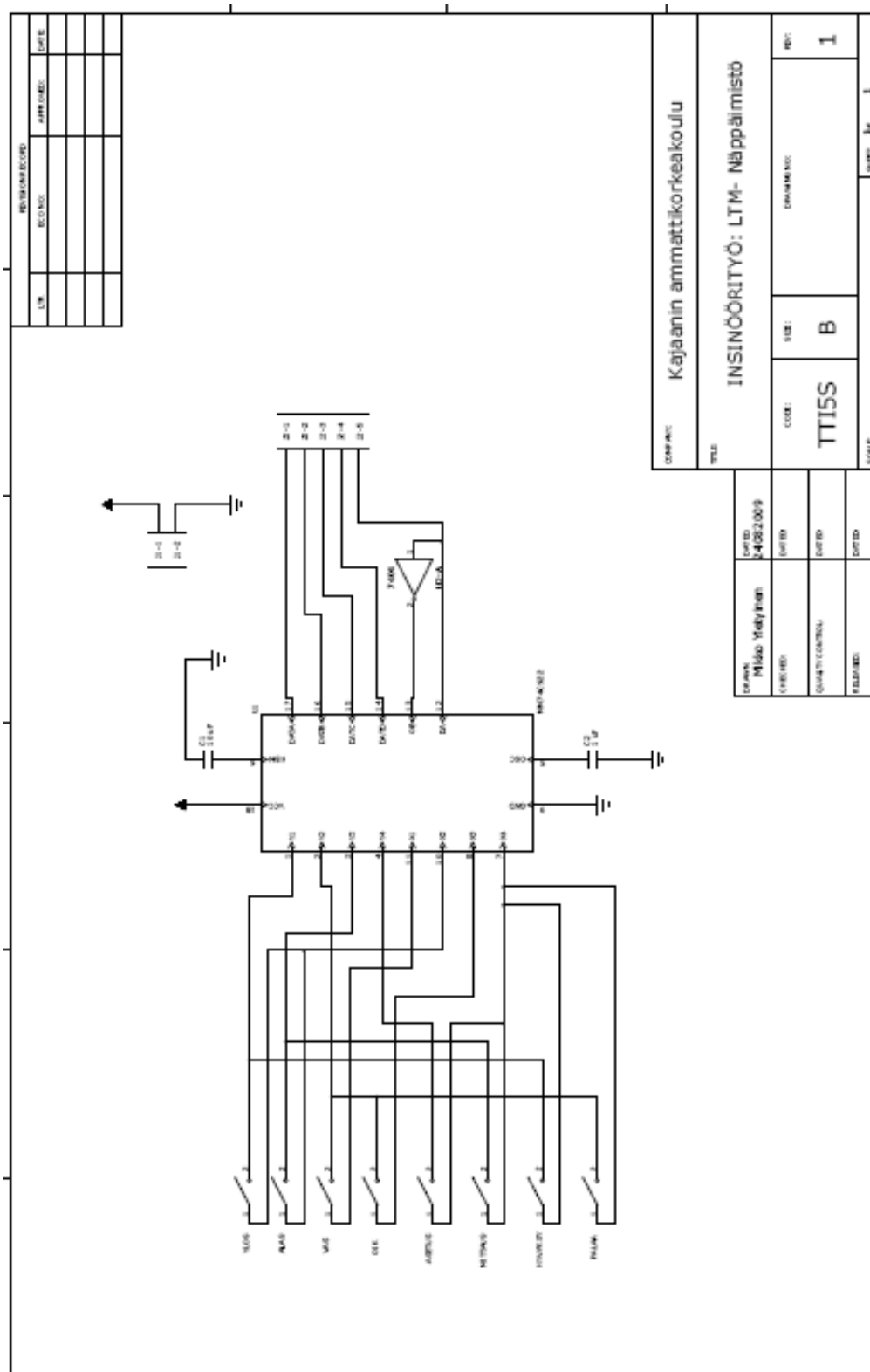




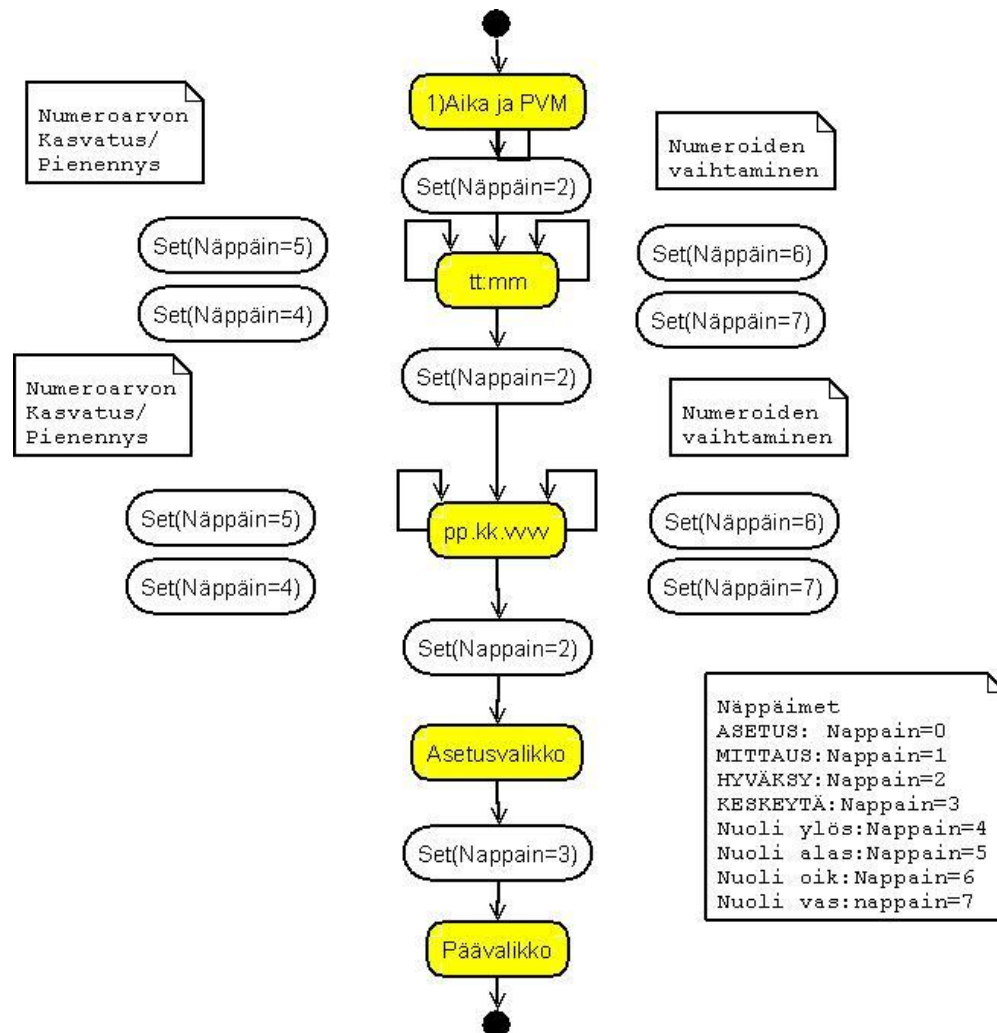


RS232-liitäntä

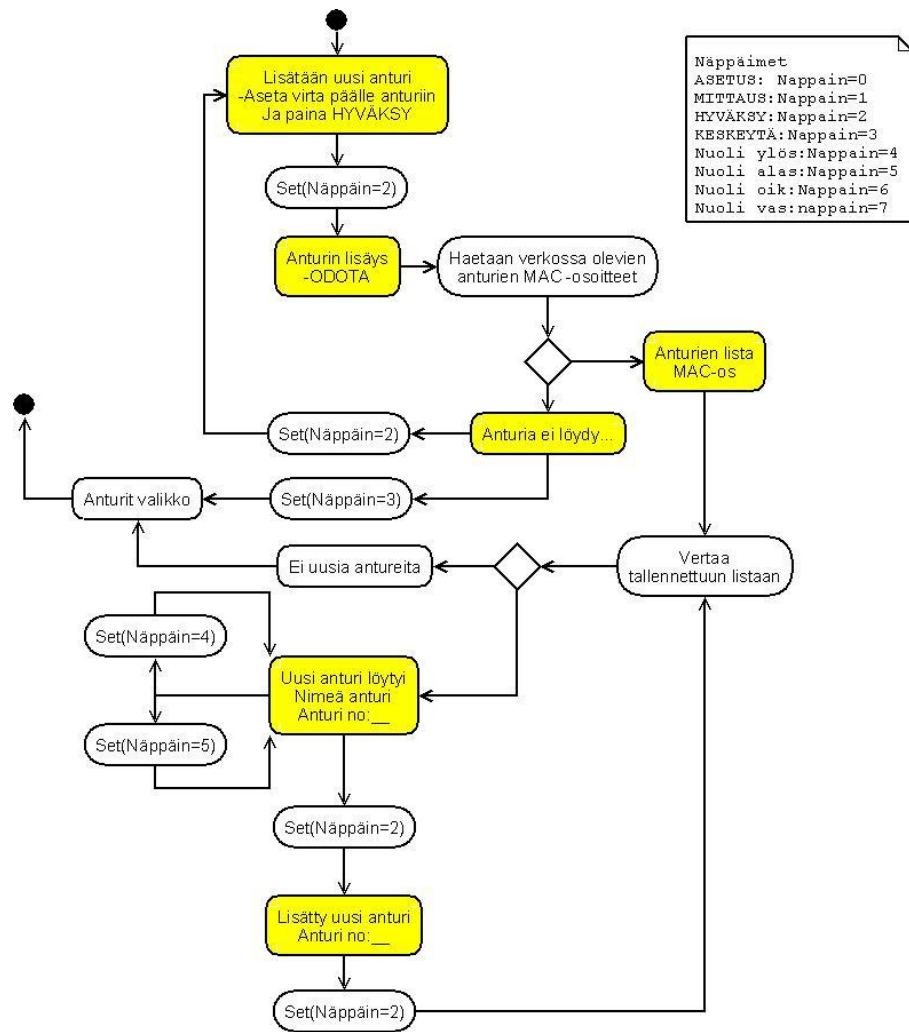
NÄPPÄIMISTÖN PIIRIKAAVIO



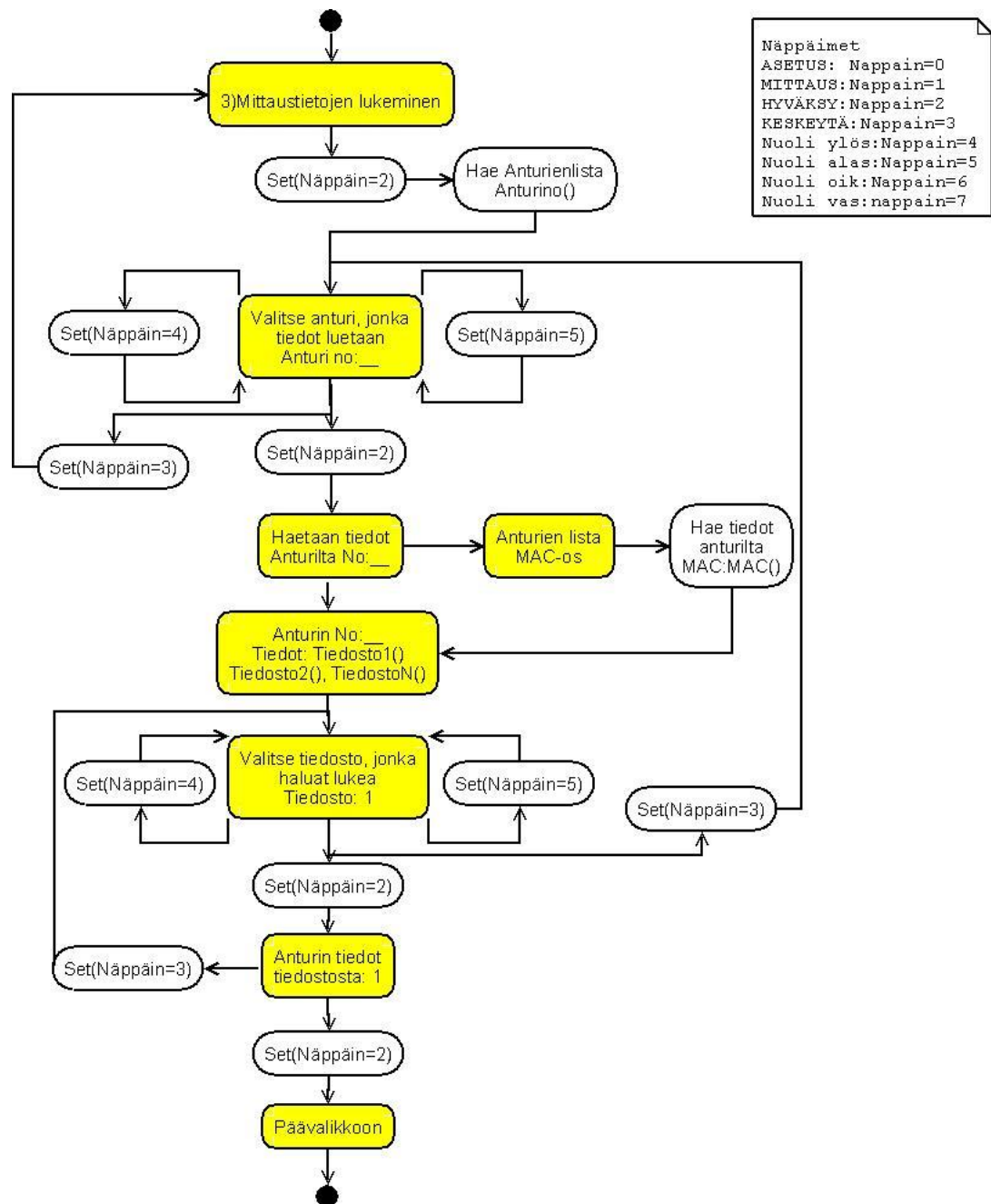
PÄÄTELAITTEEN OHJELMISTON TÄRKEIMMÄT TILAKAAVIOT



Tilakaavio 1: Kellonajan ja päivämäärän asetus



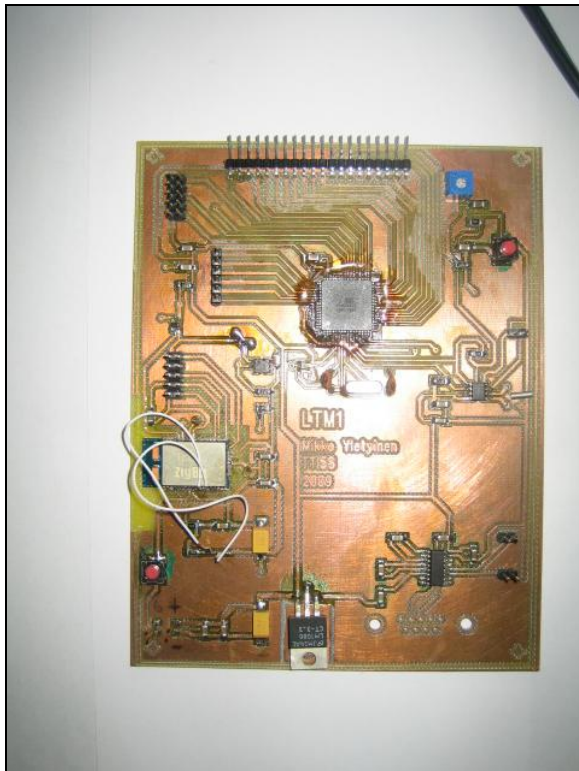
Tilakaavio 2: Anturin lisäys



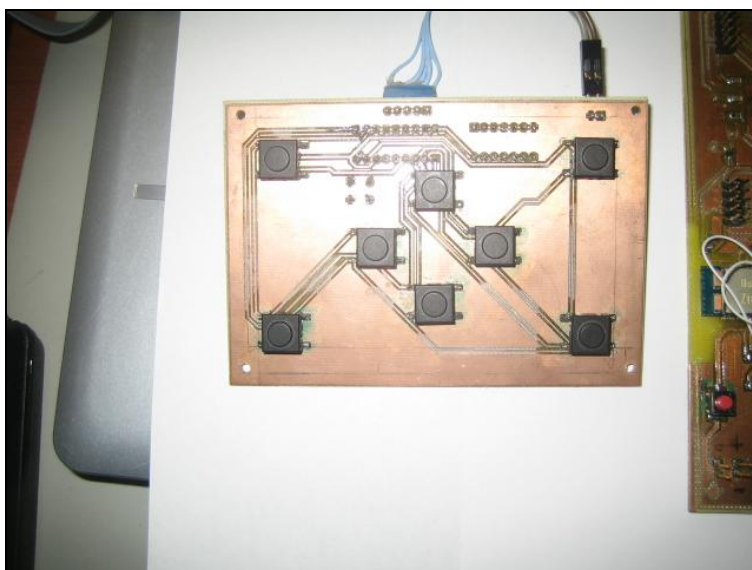
Tilakaavio 3: Mittaustietojen lukeminen anturilta

KUVIA VALMIISTA LAITTEESTA





Päätelaitteen piirilevy



Näppäimistön piirilevy

C-KIELINEN LISTAUS KELLONAJAN ASETUKSESTA

```

#include <util/delay.h>
#include "nappis.h"
#include "m41t81_ajuri.h"
#include "m41t81_apu.h"

void inittwi(void)
{
    /* Alustetaan TWI -väylän väylänopeus: 100 kHz clock, TWPS = 0 =>
    prescaler = 1 */

    #if defined(TWPS0)

        /* has prescaler (mega128 & newer) */

        TWSR = 0;
    #endif

    #if F_CPU < 3600000UL
        TWBR = 10; /* pienin TWBR arvo, kts. ATmega 128 datasheet */
    #else
        TWBR = (F_CPU / 100000UL - 16) / 2; /*TWI väylänopeus 100 kHz, RTC(M41T81) tukee
        myös 400 kHz*/
    #endif
}

void Kellon_Asetus()
{
    int tida=0; /* tida=näppäimen tila luetaan tähän muuttujaan
    int ya=0; /*UP / DOWN näppäintä osoittava
    int t,m,p,k,v; /* pikku apumuuttujat tuntien, minuuttien, päivien ja kuukausien
    indeksoimiseen
    int asetus=0; // Eri näyttötilan indeksia osoittava muuttuja
    int lippu,lippu2; //Muuttuja näytön päivitystä varten

    int tu[] = {00,01,02,03,04,05,06,07,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23};
    //Muuttuja:tunnit
    int
    mi[]={00,01,02,03,04,05,06,07,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,2
    8,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,
    58,59};
    int pa[]=
    {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31};
    int ku[]={1,2,3,4,5,6,7,8,9,10,11,12}; //Muuttujat: m=minuutit,p:päivät,k:kuukausi
    int vu[]={10,11,12,13,14,15,16,17,18,19,20,21,22,23};
    char mt[33]; //Apumuuttuja integer to ACSCII -muunnosta varten,tunnit
    char mm[33]; //Apumuuttuja integer to ACSCII -muunnosta varten, minuutit
    char mp[33]; //Apumuuttuja integer to ACSCII -muunnosta varten, päivät
    char mk[33]; //Apumuuttuja integer to ACSCII -muunnosta varten, kuukaudet
    char mv[33];

    GLCD_ClearScreen();
    // "Ohjevalikko"
    GLCD_GoTo(-64,0);
    GLCD_WriteString("KELLO JA PVM");
    GLCD_GoTo(-64,1);
    GLCD_WriteString("-----");
    GLCD_GoTo(-64,2);
    GLCD_WriteString("tt:mm:/ pp:kk:");
    GLCD_GoTo(-64,3);
    GLCD_WriteString("<- ->");
    GLCD_GoTo(-64,4);

```

```

GLCD_WriteString("*   Arvon asetus   *");
GLCD_GoTo(-64,5);
GLCD_WriteString("*           YLOS/           *");
GLCD_GoTo(-64,6);
GLCD_WriteString("*           ALAS           *");
GLCD_GoTo(-64,7);
GLCD_WriteString("*****");

```

```

while(tida !=BACK)
{

```

```

tida =LueNappis();

```

```

switch (tida)

```

```

{
case LEFT:           //Jos Vasenta on painettu

```

```

lippu=1;
lippu2=1;
if (asetus>1)
    //pienennetään asetus muuttujan arvoa
    asetus--;
else
    if (asetus<=1)
        asetus=6;
break;

```

```

case RIGHT:          //Jos oikeaa on painettu

```

```

lippu=1;
lippu2=1;
if (asetus<6) //Lisätään asetusmuuttujan arvoa
    asetus++;
else
    if (asetus>=6)
        asetus=1;
break;

```

```

default:
break;

```

```

}

```

```

if (asetus==1&&lippu==1) //Jos asetus muuttuja on yksi ja lippu on yksi:
{

```

```

switch (tida)
{

```

```

default:
if (lippu2==1)
{
GLCD_ClearScreen();
GLCD_GoTo(-64,0);
GLCD_WriteString("*   KELLO JA PVM   *");
GLCD_GoTo(-64,1);
GLCD_WriteString("*-----*");
GLCD_GoTo(-64,2);
GLCD_WriteString("*   ASETA TUNNIT:   *");
GLCD_GoTo(-64,3);
GLCD_WriteString("*           *");
GLCD_GoTo(-64,4);
GLCD_WriteString("*           *");

```

```

GLCD_GoTo(-64,5);
GLCD_WriteString("*****");
GLCD_GoTo(-64,6);
GLCD_WriteString("*****");
GLCD_GoTo(-64,7);
GLCD_WriteString("*****");
GLCD_GoTo(0,4);
GLCD_WriteString(mt);
lippu2=0;
}
break;

case UP:
if (t<23) // jos t -muutujan arvo on pienempi, kuin 23
    t++; // kasvatetaan arvoa (vielä) yhdellä
    else
        if (t>=23) // tai muuten, jos arvo on
                    //yhtäsuuri(tai suurempi), kuin 23
                    t=0; // annetaan t:lle arvo 0
itoa(tu[t],mt,10); // Muunnetaan indeksin t mukainen arvo taulukossa
//tu ASCII -muotoon

GLCD_GoTo(-64,4);
GLCD_WriteString("*****");
GLCD_GoTo(-64,5);
GLCD_WriteString("*****");
GLCD_GoTo(-64,6);
GLCD_WriteString("*****");
GLCD_GoTo(0,4);
GLCD_WriteString(mt);
break;

case DOWN:
if (t>0) // jos t -muutujan arvo on pienempi, kuin 0
    t--; // pienennetään arvoa yhdellä
    else
        if (t<=0) // tai muuten, jos arvo on
                    //yhtäsuuri(tai pienempi), kuin 23
                    t=23; // annetaan t:lle arvo 0
        itoa(tu[t],mt,10); // Muunnetaan indeksin t mukainen arvo taulukossa
//vastaava taulukon arvo ASCII
//muotoon

GLCD_GoTo(-64,4);
GLCD_WriteString("*****");
GLCD_GoTo(-64,5);
GLCD_WriteString("*****");
GLCD_GoTo(-64,6);
GLCD_WriteString("*****");
GLCD_GoTo(0,4);
GLCD_WriteString(mt);
break;

case ENTER:
lippu=1;
asetus=2;
break;

} //switch ends

} // iff ends

if (asetus==2&&lippu==1)
{
    switch (tida)
    {
        default:
        if (lippu2==1)
        {

```

```

GLCD_ClearScreen();
GLCD_GoTo(-64,0);
GLCD_WriteString("KELLO JA PVM");
GLCD_GoTo(-64,1);
GLCD_WriteString("-----");
GLCD_GoTo(-64,2);
GLCD_WriteString("ASETA MINUUTIT:");
GLCD_GoTo(-64,3);
GLCD_WriteString("");
GLCD_GoTo(-64,4);
GLCD_WriteString("");
GLCD_GoTo(-64,5);
GLCD_WriteString("");
GLCD_GoTo(-64,6);
GLCD_WriteString("");
GLCD_GoTo(-64,7);
GLCD_WriteString("*****");
GLCD_GoTo(0,4);
GLCD_WriteString(mm);
lippu2=0;
}
break;

case UP:
if (m<59) // jos m -muuttujan arvo on pienempi, kuin 59
    m++; // kasvatetaan arvoa yhdellä
else
    if (m>=59) // tai muuten, jos arvo on
        //yhtäsuuri(tai suurempi), kuin 59
        m=0; // annetaan m:lle arvo 0, Silmukka
        //pyörähtää ympäri
        itoa(mi[m],mm,10); // Muunnetaan indeksia m vastaava
        //taulukon arvo ASCII muotoon

GLCD_GoTo(-64,4);
GLCD_WriteString("");
GLCD_GoTo(-64,5);
GLCD_WriteString("");
GLCD_GoTo(-64,6);
GLCD_WriteString("");
GLCD_GoTo(0,4);
GLCD_WriteString(mm);
break;

case DOWN:

if (m>0) // jos m -muuttujan arvo on suurempi, kuin 0
    m--; // pienennetään arvoa (vielä) yhdellä
else
    if (m<=0) // tai muuten, jos arvo on
        //yhtäsuuri(tai pienempi), kuin 0
        m=59; // annetaan m:lle arvo 59
        itoa(mi[m],mm,10); // Muunnetaan indeksia m vastaava
        //taulukon arvo ASCII muotoon

GLCD_GoTo(-64,4);
GLCD_WriteString("");
GLCD_GoTo(-64,5);
GLCD_WriteString("");
GLCD_GoTo(-64,6);
GLCD_WriteString("");
GLCD_GoTo(0,4);
GLCD_WriteString(mm);

break;

/*
case ENTER:
lippu=1;
asetus=3;
break;*/

} //switch ends
//if ends

```

```

if (asetus==3&&lippu==1)
{
    switch (tida)
    {
        default:
        if (lippu2==1)
        {
            GLCD_ClearScreen();
            GLCD_GoTo(-64,0);
            GLCD_WriteString("KELLO JA PVM");
            GLCD_GoTo(-64,1);
            GLCD_WriteString("-----");
            GLCD_GoTo(-64,2);
            GLCD_WriteString("ASETA PAIVAT:");
            GLCD_GoTo(-64,3);
            GLCD_WriteString("");
            GLCD_GoTo(-64,4);
            GLCD_WriteString("");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("");
            GLCD_GoTo(-64,7);
            GLCD_WriteString("*****");
            GLCD_GoTo(0,4);
            GLCD_WriteString(mp);
            lippu2=0;
        }
        break;

        case UP:
            if (p<30) // jos p -muutujan arvo on
                p++; // kasvatetaan arvoa yhdellä
            else if (p>=30) // tai muuten, jos arvo on
                p=0; // annetaan p:lle arvo 0
            itoa(pa[p],mp,10); //Muunnetaan indeksin p mukainen arvo
            muuttujassa pa ASCII -muotoon

            GLCD_GoTo(-64,4);
            GLCD_WriteString("");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("");
            GLCD_GoTo(0,4);
            GLCD_WriteString(mp);

            break;

        case DOWN:
            if (p>0) // jos p -muutujan arvo on
                p--; // pienennetään arvoa yhdellä
            else if (p<=0) // tai muuten, jos arvo on
                p=30; // annetaan p:lle arvo 0
            itoa(pa[p],mp,10);

            GLCD_GoTo(-64,4);
            GLCD_WriteString("");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("");
            GLCD_GoTo(0,4);
            GLCD_WriteString(mp);

```

```

        break;
    }
}

if (asetus==4&&lippu==1)
{
    switch (tida)
    {
        default:
        if (lippu2==1)
        {
            GLCD_ClearScreen();
            GLCD_GoTo(-64,0);
            GLCD_WriteString("*    KELLO JA PVM    *");
            GLCD_GoTo(-64,1);
            GLCD_WriteString("*-----*");
            GLCD_GoTo(-64,2);
            GLCD_WriteString("*    ASETA KUUKAUSI: *");
            GLCD_GoTo(-64,3);
            GLCD_WriteString("*                      *");
            GLCD_GoTo(-64,4);
            GLCD_WriteString("*                      *");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("*                      *");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("*                      *");
            GLCD_GoTo(-64,7);
            GLCD_WriteString("*****");
            GLCD_GoTo(0,4);
            GLCD_WriteString(mk);
            lippu2=0;}
        break;

        case UP:
            if (k<11) // jos ve -muutujan arvo on
                    k++; // kasvatetaan arvoa yhdellä
            else // tai muuten, jos arvo on
                    if (k>=11) // annetaan arvo 0
                        k=0;
            itoa(ku[k],mk,10);

            GLCD_GoTo(-64,4);
            GLCD_WriteString("*");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("*");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("*");
            GLCD_GoTo(0,4);
            GLCD_WriteString(mk);

        break;

        case DOWN:
            if (k>0) // jos k -muutujan arvo on
                    k--; // pienennetään arvoa yhdellä
            else // tai muuten, jos arvo on
                    if (k<=0) // annetaan k:lle arvo 11
                        k=11;
            itoa(ku[k],mk,10); // Muunnetaan indeksia k vastaava
            taulukon arvo ASCII muotoon

            GLCD_GoTo(-64,4);
            GLCD_WriteString("*");

```

```

GLCD_GoTo(-64,5);
GLCD_WriteString("KELLO JA PVM");
GLCD_GoTo(-64,6);
GLCD_WriteString("ASETA VUOSI:");
GLCD_GoTo(0,4);
GLCD_WriteString(mk);

break;
}
//Switch ends
//If ends
}

if (asetus==5&&lippu==1)
{
    switch (tida)
    {
        default:
        if (lippu2==1)
        {
            GLCD_ClearScreen();
            GLCD_GoTo(-64,0);
            GLCD_WriteString("KELLO JA PVM");
            GLCD_GoTo(-64,1);
            GLCD_WriteString("-----");
            GLCD_GoTo(-64,2);
            GLCD_WriteString("ASETA VUOSI:");
            GLCD_GoTo(-64,3);
            GLCD_WriteString("");
            GLCD_GoTo(-64,4);
            GLCD_WriteString("20");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("");
            GLCD_GoTo(-64,7);
            GLCD_WriteString("*****");
            GLCD_GoTo(3,4);
            GLCD_WriteString(mv);
            lippu2=0;}
        break;

        case UP:
            if (v<11) // jos ve -muutujan arvo on
                v++; // kasvatetaan arvoa yhdellä
            else if (v>=11) // tai muuten, jos arvo on
                v=0; // annetaan arvo 0
            itoa(vu[v],mv,10);

            GLCD_GoTo(-64,4);
            GLCD_WriteString("20");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("");
            GLCD_GoTo(-64,7);
            GLCD_WriteString("*****");
            GLCD_GoTo(3,4);
            GLCD_WriteString(mv);

        break;

        case DOWN:
            if (v>0) // jos k -muutujan arvo on suurempi,
                v--; //kuin 0
            else // pienennetään arvoa yhdellä
                if (v<=0) // tai muuten, jos arvo on
                    v=11; //yhtäsuuri(tai pienempi), kuin 0
            // annetaan k:lle arvo 11
    }
}

```

```

        itoa(vu[v],mv,10);           // Muunnetaan indeksiä k vastaava
                                     //taulukon arvo ASCII muotoon

GLCD_GoTo(-64,4);
GLCD_WriteString("*          20          *");
GLCD_GoTo(-64,5);
GLCD_WriteString("*          *");
GLCD_GoTo(-64,6);
GLCD_WriteString("*          *");
GLCD_GoTo(-64,7);
GLCD_WriteString("*****");
GLCD_GoTo(3,4);
GLCD_WriteString(mv);

break;
    }                               //Switch ends
    }                               //If ends

if (asetus==6&&lippu==1)
{
    switch (tida)
    {
        default:
        if (lippu2==1)
        {
            GLCD_ClearScreen();
            GLCD_GoTo(-64,0);
            GLCD_WriteString("*KELLO JA PVM. 20 *");
            GLCD_GoTo(40,0);
            GLCD_WriteString(mv);
            GLCD_GoTo(-64,1);
            GLCD_WriteString("*-----*");
            GLCD_GoTo(-64,2);
            GLCD_WriteString("* KELLO ON: : *");
            GLCD_GoTo(21,2);
            GLCD_WriteString(mt);
            GLCD_GoTo(36,2);
            GLCD_WriteString(mm);
            GLCD_GoTo(-64,3);
            GLCD_WriteString("* PVM: . . *");
            GLCD_GoTo(21,3);
            GLCD_WriteString(mp);
            GLCD_GoTo(36,3);
            GLCD_WriteString(mk);

            GLCD_GoTo(-64,4);
            GLCD_WriteString("*          *");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("* HYVAKSY PAINAMALLA*");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("* ENTER *");
            GLCD_GoTo(-64,7);
            GLCD_WriteString("*****");
            lippu2=0;}
        break;
    }
}

```



```

        case ENTER:

            GLCD_ClearScreen();
            GLCD_GoTo(-64,0);
            GLCD_WriteString("*    KELLO JA PVM    *");
            GLCD_GoTo(-64,1);
            GLCD_WriteString("*-----*");
            GLCD_GoTo(-64,2);
            GLCD_WriteString("*    KELLONAIKA  JA    *");
            GLCD_GoTo(-64,3);
            GLCD_WriteString("*          PAIVAYS          *");
            GLCD_GoTo(-64,4);
            GLCD_WriteString("*    TALLENNETTU    *");
            GLCD_GoTo(-64,5);
            GLCD_WriteString("*                                *");
            GLCD_GoTo(-64,6);
            GLCD_WriteString("*    TAKAISIN - BACK    *");
            GLCD_GoTo(-64,7);
            GLCD_WriteString("*****");
            break;
        } //switch ends
    } //if ends

// Kun tunnrit, minuutit, päivät ja kuukaudet saadaan ylös, muunnetaan se kellon
// ymmärrettävään muotoon ja lähetetään m41t81 RTC:lle

RTC_Settime(tu[t],mi[m],pa[p],ku[k],vu[v]);

    } //While ends
} //Function ends

```